# Disease-Gene Association Using Genetic Programming

**Ashkan Entezari Heravi**

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

# Abstract

As a result of mutation in genes, which is a simple change in our DNA, we will have undesirable phenotypes which are known as genetic diseases or disorders. These small changes, which happen frequently, can have extreme results. Understanding and identifying these changes and associating these mutated genes with genetic diseases can play an important role in our health, by making us able to find better diagnosis and therapeutic strategies for these genetic diseases. As a result of years of experiments, there is a vast amount of data regarding human genome and different genetic diseases that they still need to be processed properly to extract useful information. This work is an effort to analyze some useful datasets and to apply different techniques to associate genes with genetic diseases. Two genetic diseases were studied here: Parkinson's disease and breast cancer. Using genetic programming, we analyzed the complex network around known disease genes of the aforementioned diseases, and based on that we generated a ranking for genes, based on their relevance to these diseases. In order to generate these rankings, centrality measures of all nodes in the complex network surrounding the known disease genes of the given genetic disease were calculated. Using genetic programming, all the nodes were assigned scores based on the similarity of their centrality measures to those of the known disease genes. Obtained results showed that this method is successful at finding these patterns in centrality measures and the highly ranked genes are worthy as good candidate disease genes for being studied. Using standard benchmark tests, we tested our approach against ENDEAVOUR and CIPHER - two well known disease gene ranking frameworks - and we obtained comparable results.

# Acknowledgement

I would like to express my gratitude to the following people for all their support and encouragement during the course of my research:

- *Sheridan Houghten* for her superb supervision. Her insightful and inspiring guidance shed the light on my research path and brought this work to its fullest. Thanks for your kind support and thanks for being a perfect supervisor.
- *Brian Ross* for his precious guidance and thought provoking questions. Your constant assistance will not be forgotten.
- I am grateful to my dear friend *Koosha Tahmasebipour* for sharing his brilliant ideas with me and enlightening me the first glance of research.
- Many thanks to *Cale Fairchild* for all his technical support. Your dedication is much appreciated.
- Last but not the least, I would like to thank my loving parents *Zohreh* and *Ali* and my beautiful sister *Roshanak*. Your constant care and attention made it possible for me to stay focused on my research. You are the biggest blessing in the world.

**A.E**

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The problem of associating diseases with the genes that cause them is referred in the literature by different terms, such as disease-gene association problem or disease gene prediction. It is one of the most important problems regarding human health. There is a huge effort in this field, in which different approaches are considered in order to associate genes with disease phenotypes.

Despite the great amount of work and research in this area, there is a lot more to discover, since only about 10% of human genes have been associated with disease phenotypes [4]. With the completion of human genome sequencing and with vast the amount of biological data sets, there is an unprecedented opportunity to find the genetic basis of different diseases.

In order to elucidate the inherited basis of human disease, the genomic variations should be linked to the phenotypes which can be challenging for a number of reasons. First, the concept of disease phenotype is uncertain and it is difficult to discover all of the phenotypes. This is especially because some of them can also be affected by environmental factors. The second reason is that the current techniques for genotyping are not certain and they can not provide comprehensive and reliable results for disease genes. This is one of the reasons for the emergence of methods that propose a prioritization or a set of candidate genes for disease phenotypes [29].

Despite all of these aforementioned challenges, human genetics has been successful especially for Mendelian diseases. The Online Mendelian Inheritance in Man (OMIM) [34] website says that as of August 2014, there are 5,297 phenotypes for which the molecular basis is known and 3,267 genes with phenotype-causing mutations. This success can be attributed to the available genetic tools and data sets, although as mentioned before, there is much more to discover.

## 1.1 The Problem of Disease-Gene Association and the Role of Bioinformatics

Mutated genes are the reason for many different genetic diseases or disorders. Mutation in genes occurs frequently in DNA. If the mutation is in the non-coding part of the DNA we may not notice it. However, if the mutation happens in the functional part of DNA it may have significant results and harmful consequences. SNP or *Single Nucleotide Polymorphism* is a variation occurring in the genome sequence (a single nucleotide —A, C, G or T— is mutated and replaced by another one). This is the most common reason for genetic disease. Associating diseases and their phenotypes with the genes that cause them is the effort of finding the genes for which their mutations lead to the disease. This can play an important role in human health. A better understanding of the genetic basis of human diseases can be a great help in finding better diagnosis and therapeutic strategies to treat these diseases. In bioinformatics there are different computational methods which use different available resources such as known disease genes, experiment results and other evidences in the literature (see Section 3.1), in order to associate disease phenotypes with genes.

## 1.2 Thesis Structure

The organization of this thesis is as follows. In Chapter 2 we have defined a few basic and important terms and concepts in biology, followed by the research fields in bioinformatics. One of these research fields is *gene prediction* which is the main subject of this work and is fully reviewed in Chapter 3. Complexity is one the main concepts that is used in this work. It is defined and explained in Chapter 4. The methodology, along with different databases and toolkits used in this work are described in Chapter 5. Our method has been tested on two genetic diseases: breast cancer and Parkinson's disease. These two case studies with the results and comparisons with other work are reported in Chapters 6 and 7 respectively. Chapter 8 concludes this work and discusses the obtained results, followed by the ideas for future work.

# Chapter 2

# Background

## 2.1 Bioinformatics Terms and Concepts

Bioinformatics is an interdisciplinary scientific field, which uses many areas of computer science, mathematics, statistics and engineering to process biological data. In order to work with these biological data, one should have a general knowledge of the terms and the concepts of biology. For a computer scientist who wants to work in the field of bioinformatics, getting to know these concepts can be a starting point. The aim of this chapter is to provide basic information about some important biological terms and concepts.

### 2.1.1 DNA

Deoxyribonucleic acid, or as we know it, DNA, is the unit of heredity in humans and other organisms. DNA is mostly located in the cell nucleus and if so is called nuclear DNA. A small quantity is found in mitochondria and is called mtDNA. Almost all of the DNA in body cells is the same, and even 99 percent of the DNA in different persons is the same [64].

DNA is a huge molecule in the form of a double helix. Chemically, DNA is made of three components: nitrogen-rich bases, deoxyribose sugars and phosphates. The combination of these three form a nucleotide. The huge molecules of DNA are thousands of these nucleotides which come together in pairs. The four nitrogen-rich bases in DNA can be divided into two groups: purines and pyrimidines. The purine bases are adenine (A) and guanine (G). Purine is a compound with two rings; hence, because of the chemical structure of Adenine and Guanine, they fall into this category. The other two bases are cytosine (C) and thymine (T), which because of their single

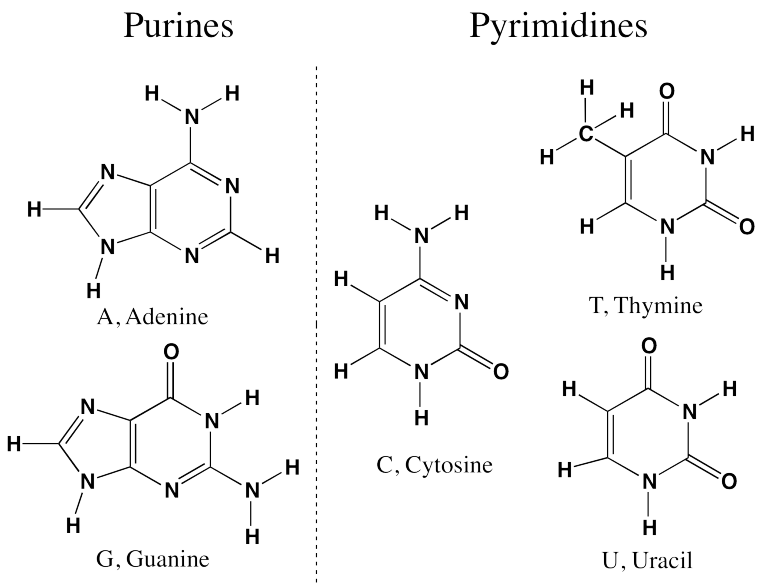


Figure 2.1: Structure of Purines and Pyrimidines. Image from [63].

six sided ring structure, are called Pyrimidines. Figure 2.1 illustrates the structure of these four bases.

Each DNA macromolecule contains thousands of these four bases. Because of their ring structure, they are flat and they can easily stack on top of each other. Therefore, the DNA molecules are compact and very strong (the reason for their extreme durability). These four bases attach to each other in pairs and they create the base pairs: adenine with thymine (A — T) and guanine with cytosine (G — C). Each of these bases is attached to a deoxyribose sugar (a ribose without oxygen atom at the 2' site) and a phosphate molecule. The whole combination, as mentioned before, is called a nucleotide. These nucleotides are arranged in long spiral strands to form the double helix. The general form of DNA is like a ladder in which the base pairs are ladder rungs and the phosphate and sugar molecules are vertical side pipes of the ladder. Figure 2.2 illustrates the form of a DNA molecule.

There are many different combinations of each of these four bases in each strand and they carry lots of information. Some portions of these strands (with a specific order of bases), code for some traits in our body which is the reason for the efforts in studying the order of these bases in DNA molecules. Note that because of the pairing in these four bases, DNA strands are antiparallel and complementary. Therefore if we have the order of bases in one strand, we can find out the base-pair sequence of the other.

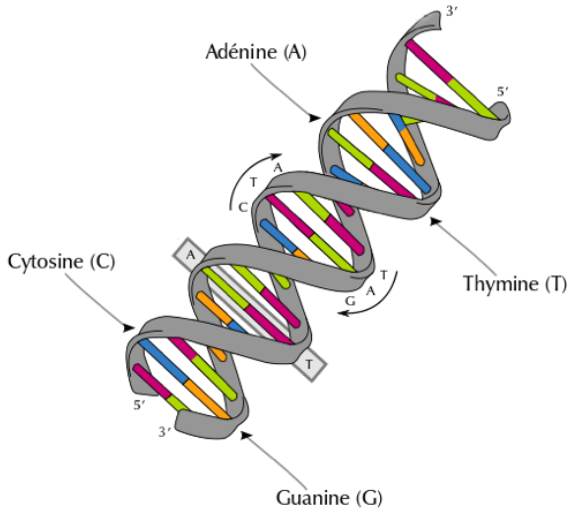


Figure 2.2: DNA structure. Image from [12].

### 2.1.2 Chromosome

A *chromosome* is a thread like structure which contains DNA and protein. The single DNA molecule in a chromosome is tightly coiled around the proteins called histones, which support its structure and control its functions. The prokaryotic cells (cells without a nucleus), usually have small circular chromosomes, while eukaryotic cells (cells with a nucleus) have larger linear chromosomes. Chromosomes can be either duplicated or unduplicated. Unduplicated chromosomes are single linear strands whereas duplicated chromosomes are two identical copies (which are called *chromatids* or *sister chromatids*) that are pinched together. The joining location of these two chromatids is called the *centromere* and the shape of a chromosome is based on the placement of the centromere (i.e. whether it is close to the middle, top or bottom of the chromosome). Refer to the Figure 2.3 to see the building blocks of chromosome.

The *ploidy number* is the number of chromosome sets held by a particular organism. For example, humans are diploids, meaning they have two copies of each chromosome. Organisms can be haploid (one copy of each chromosome), triploid (three copy of each chromosome) and so on. In humans, as diploids, there are 22 pairs of uniquely shaped autosomal (non-sex) chromosomes and one pair of sex chromosomes (those chromosomes that determine our gender).

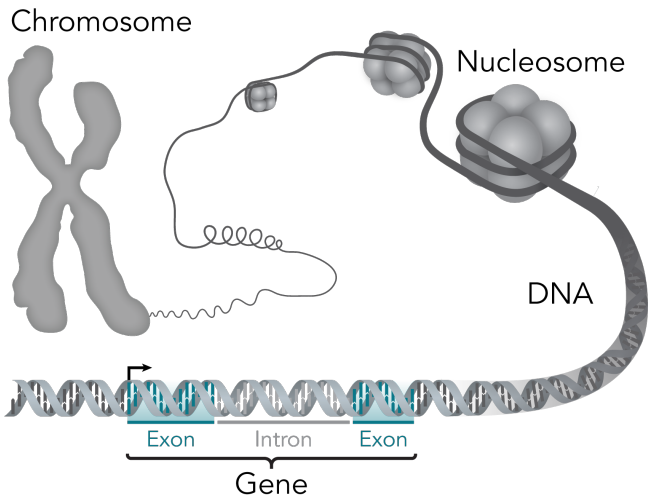


Figure 2.3: Structure of Chromosome. Image from [95].

Chromosomes carry *genes* (Figure 2.3). Genes are sections of DNA molecules that code for particular observed traits, called *phenotypes*. To pass these genetic traits from one generation to the next, the chromosomes must be replicated. Each pair of homologous chromosomes carries the same, but not necessarily identical genes which are called *alleles*. Each gene can have one or more alleles. The combination of alleles of all the various genes is called the *genotype*.

### 2.1.3 RNA

Ribonucleic acid, or RNA, is a macromolecule like DNA which is present in all living cells. RNA is very similar to DNA in many aspects. Unlike DNA, RNA is single stranded. The reason is that it is using ribose sugar instead of deoxyribose (deoxyribose is a ribose which lacks oxygen on 2' location). The four nucleotide bases in RNA are adenine, guanine, cytosine, and uracil (instead of the thymine in DNA). A RNA strand, has a backbone of alternating sugar (the ribose mentioned earlier) and phosphate groups (see Figure 2.6). Unlike DNA, RNA molecules are very unstable and they decompose very rapidly. Sometimes, complementary regions bond together and form a secondary structure. These structures can be in different forms such as stem-loop (Figure 2.4), internal loop (Figure 2.5) or pseudoknot. These *secondary structures* can be important in the function of the RNA molecule. There are different works that attempt to predict these structures such as CyloFold [33], KineFold [46] and RNAfold [38].

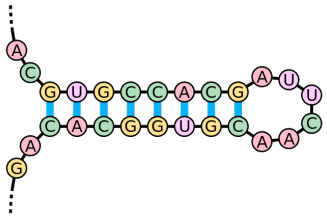

Figure 2.4: RNA stem-loop. It happens when there are complementary regions on the same strand. Image from [15].

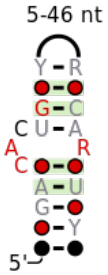


Figure 2.5: Example of internal loop in RNA strand. Image from [14].

There are different types of RNA in the cell which play an important role by catalyzing biological reactions. These different types are messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA). mRNA regulates the expression of genes, tRNA is for carrying the amino acids (building blocks of proteins) during translation of genetic language of nucleic acid to the language of protein (the process for generating the proteins, based on DNA genes). rRNA is for attaching the amino acids in chains, during the process of aforementioned translation. In eukaryotic cells, DNA is a vital part of the cell and it never leaves the cell. Therefore, RNA will act like a copy of the DNA in the cell and during a process of transcription (copying DNA's message into RNA's language), messages of genes will be transcribed into RNA (mRNA). This transcribed RNA will leave the cell and it then goes through a process which ends in generating proteins [65].

### 2.1.4 Proteins

Proteins are biological macromolecules which play many critical roles in the body. Proteins are made of amino acids and in each protein molecule, there are hundreds to thousands of amino acids, attached together. Since there are 20 different amino

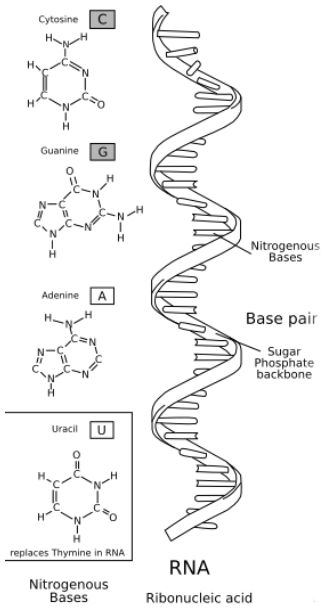


Figure 2.6: Structure of RNA molecule. Image from [16].

acids and because each molecule of proteins (polypeptides) usually has more than 100 amino acids, there is a huge variety of combinations. Proteins perform many important tasks within cells and they are required for the structure, function and regulation of different tissues and organs in the body. Proteins can be divided into different groups, according to their function. In Table 2.1 you will see a list of different tasks that they perform in the body.

The main difference between proteins is their amino acid sequences which cause the unique structure and function of different proteins. Determining the 3D structure of proteins is an important issue in different fields such as bioinformatics. It will help scientists to understand the functions of proteins at a molecular level. There are four distinct levels of protein structure:

1- *Primary Structure*: the primary structure of protein refers to their linear sequence of amino acids. This primary structure is held together by covalent or peptide bonds.

2- *Secondary Structure*: This structure refers to highly regular local sub-structures. The two main types of this structure are alpha helix and beta strand which are held together by hydrogen bonds.

3- *Tertiary Structure*: tertiary structure refers to the three-dimensional structure of a protein molecule. The protein folding (which affects its structure) is self

Table 2.1: Protein Functions

| Function | Description | Example |
|---|---|---|
| *Antibody* | Antibodies bind to specific foreign particles, such as viruses and bacteria, to help protect the body. | Immunoglobulin G (IgG) |
| *Enzyme* | Enzymes carry out almost all of the thousands of chemical reactions that take place in cells. They also assist with the formation of new molecules by reading the genetic information stored in DNA. | Phenylalanine hydroxylase |
| *Messenger* | Messenger proteins, such as some types of hormones, transmit signals to coordinate biological processes between different cells, tissues, and organs. | Growth hormone |
| *Structural component* | These proteins provide structure and support for cells. On a larger scale, they also allow the body to move. | Actin |
| *Transport/storage* | These proteins bind and carry atoms and small molecules within cells and throughout the body. | Ferritin |

organizing and is highly affected by the amino acid sequence (primary structure) of that protein. However, the environment can also affect the folding of a protein and its final shape.

4- *Quaternary Structure*: if the protein is built of sub-units, the quaternary structure is concerned with how these subunits fit together.

Figure 2.7 illustrates the four levels of protein structure.

As it is explained in the previous sections, there is a natural flow of sequential information from DNA to RNA, and from RNA to protein. This detailed transfer of sequential information is known as the central dogma of microbiology [17] which states that these data can not be transferred back from protein to another protein or to a nucleic acid.

## 2.2 Research Fields in Bioinformatics

There is a huge amount of data in biology which needs to be organized and processed. The two main efforts in bioinformatics are storage and analysis of data. Bioinformaticians try to construct a good infrastructure for storing data which will make it easy to process the data and analyzing it. Most of the data is in the form of sequences of nucleotides in DNA, or sequences of amino acids in protein chains. These sequences

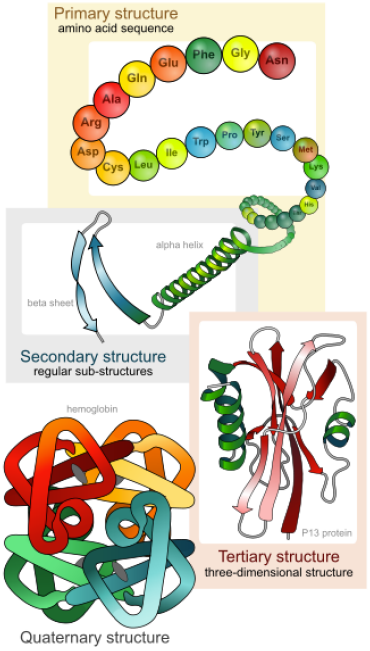


Figure 2.7: Four levels of protein structure. Image from [13].

can affect the structure of their molecules. Bioinformaticians study these sequences and try to determine the final structure of these molecules. If they know these structures, they can deduce different features of them. Another important issue in this field is trying to find the genes in DNA sequences. Genes are coding parts of DNA (exon) which code for something (i.e. they have the instructions for producing specific proteins). Bioinformaticians work on these sequences to determine these genes and their effects. Also there are other interesting topics such as interaction between proteins (protein-protein interactions) or relationships between different genes (polygenic trait, a trait which is controlled by multiple genes). In the following sections, a summary of some of the most important research topics in bioinformatics is provided. It should be noted that bioinformatics is far broader than the following topics. For more information refer to [32].

### 2.2.1 Sequencing

In sequencing, the aim is to determine the primary structure of a biopolymer (like the order of nucleotides in DNA). The result of the sequencing is a linear order of the elements which represents a great deal of information about the sequenced molecule. These results are among the most important data in bioinformatics databases and can be used and analyzed for different purposes. In genetics, different types of sequencing

are DNA sequencing, RNA sequencing, protein sequencing, and polysaccharide sequencing. DNA sequencing has great importance and there are different methods for sequencing DNA. The most commonly used method historically, is Sanger sequencing [78] (the chain termination method), but there are other methods like pyrosequencing. In DNA sequencing, the DNA is first fractured into millions of small pieces. Then, these small pieces will be read and after that they should be put together. This process which is known as genome assembly has its own difficulties. Different methods exist for merging the small pieces of DNA.

### 2.2.2 Sequence Alignment

As sequencing technologies grow, there will be more data available each day. One of the central challenges to analyze this data is sequence alignment. The aim of sequence alignment is to arrange the sequences in such a way that areas of similarities can be found. These similar areas can show some functional, structural or revolutionary relations between the sequences. There are different possible ways of aligning two sequences, assuming we can use gaps in these alignments, which means an insertion of a letter in one sequence or deletion of a letter in the other sequence. The two most important ways of aligning two sequences are local and global alignments. In local alignment, it will be assumed that the query is a portion (substring) of the other string, so the query will be matched with a substring of the main string. Whereas in global alignment, an end to end alignment will be performed and there will be many gaps in the final alignment, if the size of the two strings are dissimilar. The Needleman-Wunsch algorithm is a general global alignment algorithm and the Smith-Waterman algorithm is general local alignment method [51].

### 2.2.3 Gene Prediction

DNA sequences have two main parts: *exon* and *intron*. The exon is the part of the sequence that codes for something (which is called coding DNA or gene) and the intron is the part that does not code for anything (sometimes called "junk DNA"). While working with the raw data of sequences, one of the main efforts is to find the genes. Gene prediction is the process of finding the genes in sequences. It should be distinguished with finding the function of a gene which is another important research area with various methods. There are different methods for finding the genes. Some are empirical methods, which try to find the known expressed sequence tags, mRNA and protein products in the target genome. If the sequence is mRNA, the problem

is only to find a gene sequence in that mRNA. If the given sequence is a protein sequence, after reverse translating, a variety of possible candidates for coding DNA sequences can be derived. Then the problem is trivial to search the given sequence to find a match for a coding DNA (using different methods such as local sequence alignment). Another method for finding the genes are ab initio methods. They look up the sequence to find signals and evidence of existence of a protein coding gene.

# Chapter 3

# Literature Review of Computational Methods

## 3.1 Computational Methods

In order to find the genes related to a genetic disease, there will be a search space in which the methods look for the genes causing the diseases. The results of these methods are usually a group of genes which are called *candidate disease genes* and have a high probability of being involved in the genetic disease. Generally there are two main approaches to represent the candidate genes: *prioritization* or *selection*. In prioritization, genes will be ranked (or assigned values) based on the likelihood of their involvement, and in selection, a reduced subset of genes with a higher likelihood of involvement will be selected as the candidate disease genes [73].

One of the ways of determining the candidate genes is *linkage analysis*, which is the study of the tendency of genes to be inherited together because of their close location to each other on the chromosome [66]. Another recent methodology for disease gene prediction is *genome wide association studies* (GWAS), an approach which scans the markers across the complete sets of DNA or genomes in order to find genomic variations and SNPs associated with a particular disease [59]. A computational disease gene prediction is a method which makes use of huge genomic databases and repositories of biomedical literature to identify the most likely disease gene candidates. These candidate genes can be further studied by researchers for empirical analysis in order to find diagnose, prognose and therapeutic treatments for genetic diseases [87]. In the next few sections, different evidence types that are used by different computational methods are explained and reviewed.

### 3.1.1 Text Mining of Biomedical Literature

There are several different existing computational disease gene prediction methods in the literature, one of which is text mining of the literature. PubMed [1] and OMIM [34] are the two approaches based on natural language processing and text mining of the biomedical literature [93]. It should be considered that although these data resources offer worthwhile knowledge, they are inherently biased toward well-studied genes [73].

### 3.1.2 Functional Annotations

Another group of disease gene prediction methods make use of functional annotations in which they try to find the genes in the same pathway. These biological pathways are a series of events which are the result of interactions within cells which lead to a specific phenotype. In these methods, genes with the same functionality will be considered as a candidate disease gene. The Kyoto Encyclopedia of Genes and Genomes (KEGG) [39], Gene Ontology (GO) [6] and Human Phenotype Ontology [43] are three significant databases which connect genes with their functionality.

### 3.1.3 Gene Properties and Sequencing Data

Inherent gene or protein properties such as length, phylogenetic analysis of genes, degree of conversation, and next generation sequencing data can be used for associating genes with diseases. It should be noted that although these approaches can be successful, due to lack of knowledge about the aforementioned information (especially gene sequences), they are not powerful enough to find many genetic causes of diseases. The relationships between phenotypes of candidate genes with known disease genes is another way of finding new candidate genes. Some data resources such as MimMiner [89] and OMIM [34] can be used to connect genes with phenotypes.

### 3.1.4 Gene Expression

Data from gene expression is another useful resource which can be helpful toward finding candidate genes. There are different factors such as cell type and environment that affect the expression of the genes. It is also obvious that the expression of disease

genes is different than the normal ones. The methods which use gene expression data compare the expression of the disease genes and normal genes. These methods are capable of making a connection between the disease genes and new candidate genes based on their similarity of expression on microarrays. It should be noted that the data produced in this way is one of the least biased regarding disease gene prediction [73].

### 3.1.5 Protein-Protein Interaction

One of the most effective and powerful tools for predicting disease genes is Protein-Protein Interaction networks or PPI [60]. In a PPI whose nodes are proteins, and the links represent interactions between them, it has been shown that the genes related to the same disease have protein products which physically interact [31]. Based on this, these methods analyze the PPI networks in order to find proteins which physically interact and then consider their producing genes as candidate disease genes.

At this point, it should be mentioned that there have been some efforts regarding disease gene association which try to take advantage of several different methods mentioned above. In this case there can be some problems with combining the ideas which is mainly concerning how to combine all of them together as an integrated computational module. If planned appropriately, these fusion methods are expected to produce insightful results. The following section provides an overview of some of the important computational methods.

## 3.2 Using a Fusion of Evidence Types in Different Computational Methods

As mentioned earlier, there are some efforts in different computational methods to take advantage of different evidence types. Using a fusion of evidence types has its own challenges. For example which evidence types to use or how to combine and integrate them into a single operational module. Also how to combine each generated value of different evidence types into a final result is another challenge in here.

One of the methods that combines different evidence types is CIPHER [94]. In this computational framework, in order to find the relations between phenotypes and genotypes, they are looking at the similarities between disease phenotypes, known

Table 3.1: Using a fusion of evidence types in some well-known disease-gene association methods

| Method | Evidence Type | Reference |
|---|---|---|
| CAESAR | functional annotation, gene expression, inherent properties, text mining | [27] |
| MAESTRO | functional annotation, gene expression, inherent properties, text mining, PPI | [8] |
| GFINDER | gene expression, phenotype relationship | [55] |
| GENESEEKER | functional annotation, gene expression, text mining, phenotype relationship | [88] |
| CGPRIO | functional annotation, inherent properties, text mining, PPI | [26] |

associations between genes and phenotypes and also the interactions among proteins. This is based on the assumption that functionally related genes can cause similar diseases.

ENDEAVOUR [2] is another framework that provides gene prioritization through fusion of genomic data. This prioritization is based on the similarity to the known disease genes involved in these phenomena. ENDEAVOUR creates multiple distinct prioritizations for heterogeneous data sources and then using order statistics, generates a global ranking. ENDEAVOUR has the flexibility for adding additional data sources. Its accessible online tool [21] provides different evidence types such as PPI, text mining and gene expression.

GPEC [50] is also a gene prioritization method that finds genes that are likely to be associated with a disease or involved in a pathway. GPEC is a network based approach that uses a random walk with restart algorithm (RWRA). It uses protein-protein interaction network and functional annotations to find the candidate genes. There are other frameworks that use a fusion of evidence types. Table 3.1 lists a few of these methods.

# Chapter 4

# Complexity

The highly interconnected nature of the human interactome (see Section 4.2) makes it difficult to consider single genes for diseases. In fact, these myriad number of interactions will be processed in order to find functional and topological modules of genes. Further studies showed that the effects of a genetic abnormality is not only restricted to the genes that carry it, but it can spread along the network in which these genes interact and affect the gene products of the normal genes as well [7]. Furthermore, it has been revealed that the same situation exists with the phenotypic impacts of the mutated genes that their phenotypes are a result of the mutated genes as well as the genes and gene products with which they interact [7]. Based on these facts, the importance of the network based approaches in disease gene association can be understood and the question would be "what types of networks should be studied?".

There are many organs in human body which interact in different ways. In the past decade, there has been a huge growth in the amount of human specific interaction data [35] which leaves us with different networks that can be studied. Among these networks, protein-protein interaction networks or PPI are among the most important networks to be studied [60]. There is also metabolic networks with the nodes representing metabolites and the links representing participation in the same biochemical reactions [7]. Other important networks are regulatory networks, RNA networks, co-expression networks and genetic networks [7].

## 4.1 Complex Networks

Considering the biological networks in the human body, it should be noted that these networks are extremely huge with thousands of nodes and links among them.

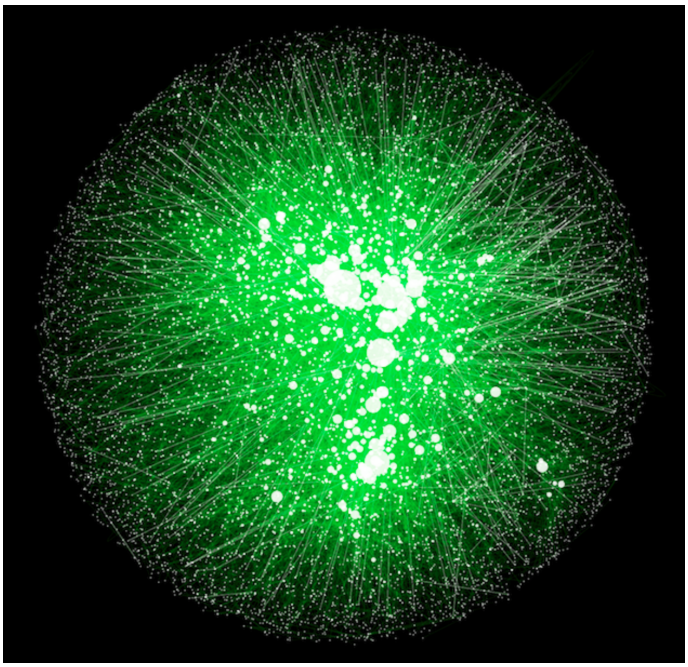

Figure 4.1: Human Interactome. Image from [18].

Refer to Figure 4.1 for an example. These networks have complex behaviour and they are called *complex networks*. Many real world systems are complex networks, including biological networks, transport networks, the Internet, and the World Wide Web. When it comes to the complexity of these networks, the classic algorithms for regular graphs will no longer work in a reasonable time, therefore these complex networks should be studied and new algorithms should be used in order to process these networks.

There are many common properties in complex networks such as small world phenomena or community structure. These complex networks and their properties need to be analyzed in order to extract different information.

## 4.2 The Modularity of Genetic Diseases

When it comes to the problem of disease gene association, there are several concepts related to the modular nature of genetic diseases that should be considered and it should be noted that almost all of the computational methods use a network based approach in order to find candidate genes. One of these important concepts is that there is not a one-to-one relationship between genes and genetic diseases or disorders. This means that one gene can take part in several phenotypes of different diseases,

and one disease can be a result of mutation in different genes [70]. There is a huge network known as the *human interactome* which is a complex network of communications between different cells and organs. This network is extremely huge and has around 25,000 protein coding genes, around a thousand metabolites and an undefined number of proteins and RNA molecules. Nodes of this network are different cellular components, and number more than 100,000. The links between these nodes, which represent interactions between these components, are expected to be much larger in number. Observing this huge network and excessive amount of interactions between the genes and other components, it can be simply inferred that genetic diseases are not the result of a single gene, but they are a result of different genes, closely interacting with each other as modules in a complex network [7].

### 4.2.1 Guilt by Association

The principle known as *guilt by association* says that the genes interacting closely with each other tend to share the same functionality, or from another point of view, it can be said that the genes involved in a disease will closely interact with each other in their underlying networks [30]. Based on this principle, the computational methods for disease gene association take advantage of known disease genes (the genes that are already known to be involved in the given genetic disease). First they will identify the set of already known disease genes and then try to find the physical modules based on the location and closeness in the network, or functional modules which contain these known disease genes. They will then use this information to select the genes in these modules as new candidate disease genes. These candidate disease genes are the genes that are expected to be involved in a given genetic disease.

## 4.3 Centrality Measures

Centrality measures are important values for the nodes of a network which reveal the key nodes that have significant roles in the network, relevant to the network topology [24, 80]. In this section, several centrality measures are defined. For this, we assume that we have a graph $G = (V, E)$ in which:

- $n$ is number of vertices and $deg(v)$ is the degree of vertex $v$.
- $dist(v, w)$ is the shortest path between $v$ and $w$.
- $\sigma_{st}$ indicates the number of shortest paths between $s$ and $t$, and

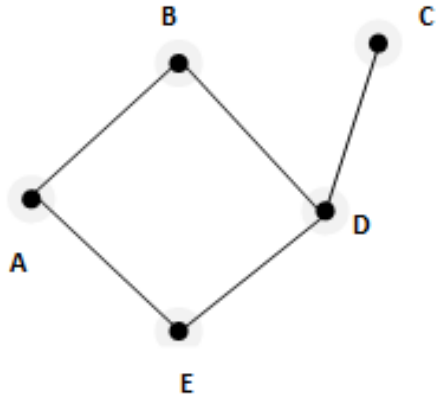


Figure 4.2: Graph G with 5 vertices and 5 edges.

- $\sigma_{st}(v)$ is the number of shortest paths between $s$ and $t$ that pass through vertex $v$.

The following explanations are based on the supplementary data of [80].

### 4.3.1 Degree

The *degree* of node $v$ or $deg(v)$ is simply the number of nodes adjacent to $v$. By adjacent we mean those nodes that are directly connected or first neighbours, hence it corresponds to adjacent edges as well. For example, in graph G (Figure 4.2), vertex D has the highest degree which is 3. Vertices A, B and E all have a degree of 2 and vertex C has a degree of 1.

One of the applications of degree is calculating the degree distribution $P(k)$ which is the probability that a node has exactly $k$ links. $P(k)$ can be calculated by counting the number of nodes with $k$ links, for $k = 1, 2, 3, ...$ and then dividing it by the total number of nodes which is $n$. Based on the degree distribution decisions can be made about the structure of network. For example, a graph with *power law* degree distribution indicates the existence of a few nodes with high degree (hubs). In biological networks, a node with a high degree can suggest that it is a node with a central regulatory role. For example proteins with high degree in signaling networks can suggest that they are regulatory hubs of the network.

### 4.3.2 Diameter

Diameter is defined as the longest distance between any pair of vertices of a graph. Note that distance in here is the shortest path between the corresponding vertices. In graph G of Figure 4.2, diameter is 3, which is the shortest path between vertex C and vertex A. Note that there are two paths between vertex E and D, but we always consider the shortest path. Therefore, $dist(E, D) = 1$ and $dist(A, C) = 3$.

This centrality measure indicates the distance between the two most distant nodes of the network, hence, a high diameter in graph can imply low graph compactness. However it should be considered that this doesn't always work, as in a network that has two distant nodes but other nodes are located close to each other. In this example, although the diameter is high, the graph is still compact. In contrast with high graph diameters, a low graph diameter is much more informative and implies high compactness in the graph. It should be noted that in order to decide that the diameter is low or high, we should consider the total number of nodes in the graph.

In biological networks, a low diameter, which implies compactness, can be a proof of close interactions between nodes. For example, in a compact PPI (protein-protein interaction) network, based on the principle of guilt by association (see 4.2.1), shared functionalities can be implied.

### 4.3.3 Average Distance

This parameter is defined as the average of all shortest paths in a graph. This is determined by dividing the summation of all shortest paths (between any pair of vertices) by the total number of nodes.

$$AvD_G = \frac{\sum_{v \in V} \sum_{w \neq v \in V} dist(v, w)}{n} \tag{4.1}$$

As with diameter, average distance can be used to imply graph compactness. However, since average distance considers all nodes and shortest paths, the average distance is more reliable than graph diameter. A high average distance implies a sparse graph and vice versa. Again, in determining low or high, the size of the network (total number of nodes) should be considered. In a large biological network, a low average distance shows closeness and easiness of communications and indicates existence of functional complexes and modules.

### 4.3.4 Eigenvector

The eigenvector is a centrality measure that can be defined recursively. A node will have high eigenvector value, if it has many neighbours with high eigenvector values. The concept is to find nodes that not only have a high number of neighbours, but also have important (high scoring) neighbours. Based on this, a node with high eigenvector value can be said to be well-connected and that node will be visited frequently while traversing the graph. Equation 4.2 shows one way to calculate eigenvector.

$$Eig(v) = \frac{1}{\lambda} \sum_{w \in N(v)} Eig(w) \tag{4.2}$$

In this equation, $\lambda$ is a constant and $N(v)$ is the set of the neighbours of vertex $v$. This recursive formula can begin by giving an initial eigen value of 1 to every node, and it repeats until the values stop changing.

In biological networks, the eigenvector value can be an informative value for the regulatory role of nodes in that network. A node with a high eigenvector value, has several neighbours with high eigenvector values which regulate them or are regulated by them. On the other hand, a node with a low Eigenvector value is a peripheral node, and this node is not interacting with central nodes.

### 4.3.5 Eccentricity

Eccentricity, which is a node centrality index, is calculated by finding the shortest path between the node $v$ and all other nodes in the graph. The reciprocal of the maximum value will be reported as the eccentricity value:

$$C_{ecc}(v) = \frac{1}{max\{dist(v, w) : w \in V\}} \tag{4.3}$$

Since we are finding the most distant node from a node, we can make assumptions about node proximity and centrality role. In this sense, a high value for eccentricity means that all nodes of the network are in proximity of that node, but a low value for eccentricity means that other nodes are far from this node (or at least there is one distant node from this node). Therefore, a high eccentricity value is much more informative than a low value.

In biological networks, eccentricity of a node can be compared with the average eccentricity of the network, a higher value indicates the easiness of that node to be reached and influenced by other nodes of the network. A low value of eccentricity

(compared to the average) can imply a marginal role for that node.

### 4.3.6 Closeness

Closeness is a centrality index that is computed by finding the reciprocal of summation of shortest paths from a given node to all other nodes of that network. The closeness of node $v$ is calculated by the following formula:

$$C_{clo}(v) = \frac{1}{\Sigma_{w \in V} dist(v, w)} \tag{4.4}$$

Based on this formula, a high value of closeness for a node shows the node proximity in the network. Similarly, a low value of closeness implies that other nodes of the network are distant from that node. While using this measure, it should be noted that this value can be misleading. For example, a low value of closeness that can be obtained by a few very distant nodes, while other nodes are close, and the same thing can happen with a high closeness value. Therefore, this value is not very informative by itself, but combined with other measures can be much more reliable and informative. For example, a high eccentricity value along with a high closeness value can imply that the node is central in the network.

If a node in a biological network is close to other nodes and has a higher closeness value than the average of other nodes, this can imply that this node is functionally relevant to other nodes. Also, nodes with low closeness can be of interest, since it shows that these nodes are less relevant to this network and possibly they act as intersecting boundaries with other networks. Based on this, if a biological network has a high average closeness, it can be a sign of functional modules existing in that network.

### 4.3.7 Radiality

Radiality is a centrality index with respect to graph diameter. In order to calculate this value for a node $v$, summation of shortest paths from $v$ to all other vertices will be subtracted by graph diameter +1 ($\Delta_G + 1$). Then this value will be divided by $(n - 1)$.

$$C_{rad}(v) = \frac{\Sigma_{w \in V}(\Delta_G + 1 - dist(v, w))}{n - 1} \tag{4.5}$$

Since graph diameter ($\Delta_G$) is the longest shortest distance, subtracting the distance from $v$ to its neighbour nodes from diameter gives a high value to the radiality of that

node if the paths are short and vice versa. Therefore a high radiality value can imply the proximity of that node, and if the radiality is low, that node would be peripheral. Combining radiality with closeness and eccentricity can lead us to meaningful results in terms of node centrality role: a high value for radiality, closeness and eccentricity is a consistent sign that the node occupies a central position in the network. In biological networks, a high radiality value compared to the average of graph, indicates the existence of functional modules which can be helpful in obtaining the unknown functionality of genes based on the functionality of close genes.

### 4.3.8 Centroid Value

Centroid value is the most complex centrality measure which is calculated by the following formula:

$$C_{cen}(v) = min\{f(v,w) : w \in V\} \tag{4.6}$$

where $f(v,w) = \gamma_v(w) - \gamma_w(v)$ and $\gamma_v(w)$ is the number of vertices that are closer to $v$ than to $w$. With this function, the number of nodes that are closer to $v$ or $w$ (in terms of shortest path) is calculated and compared the node distance of all the other nodes from the others. This means that this function is comparing the number of neighbours (not necessarily immediate neighbours) between pairs of nodes. In this sense, a high value for centroid score indicates that the node has a higher number of neighbours (not only first neighbours). Again, here the centroid value is more meaningful when it is compared with the average centroid value of all the nodes in the graph. In a biological network, such as a protein signaling network, a high value of centroid can imply the capability of a protein to functionally organize discrete modules of proteins. Based on this, a network with a high average centroid value can be organizing functional units.

### 4.3.9 Stress

This centrality measure is about the number of shortest paths passing through a specific node. In order to calculate the stress of a node, all shortest paths in the graph will be calculated and then the number of shortest paths passing through a node $v$ will be reported as its stress:

$$C_{str}(v) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \sigma_{st}(v) \tag{4.7}$$

A stressed node (a node with a high value for stress) is a node included in many shortest paths which indicates an important role in the network. However, this doesn't mean that this node is crucial for those connections, because there can be other shortest paths not passing through this node. Again this stress value is more informative when it is compared with the average stress value of the graph. In biological networks, a stressed node shows the capability of holding communications between nodes. Because of the nature of this measure, a stressed node may not be relevant in holding the connections but it can be heavily involved in cellular processes.

### 4.3.10 Betweenness

Betweenness is a node centrality index similar to stress but much more informative. In order to calculate betweenness, pairs of nodes $s$ and $p$ are considered, then the number of shortest paths between them passing through a node $v$ is divided by the total number of shortest paths between them:

$$C_{spb}(v) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \delta_{st}(v) \tag{4.8}$$

where

$$\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{4.9}$$

If the number of shortest paths between $s$ and $p$ is equal to (or close to) the number of shortest paths between $s$ and $p$ that pass through $v$, the value of $\delta_{st}(v)$, and therefore the value of betweenness, will be higher. This shows the importance of the node $v$ in maintaining the connections between other nodes. Therefore a high betweenness score indicates the importance of that node to maintain the connections, and looking at the stress value, the number of paths for which the node $v$ is critical will be known. Hence, betweenness and stress can be used together to obtain complementary information. In biological networks, a high value of betweenness shows the capability of that node in holding the communications of other nodes and organizing the regulatory modules.

# Chapter 5

# Methodology

Our approach to the disease-gene association problem combines several ideas. In short this approach analyzes the complex network around the already known disease genes and extracts several centrality measures for every node in this network. Based on these measures, a GP program has been developed which aims at finding a pattern in the known disease genes. It then tries to find this pattern in other genes of this network and based on this, it will rank the genes. The outcome of this approach is a ranking which prioritizes the genes based on the likelihood of their involvement in the given genetic disease. In the following sections, an overview of the methods and platforms used in this work is presented.

## 5.1 Genetic Programming

*Artificial intelligence* and *machine learning* try to bring the intelligence to machines and software and give them the problem solving skills. As Arthur Samuel, one of the pioneers in this field stated, the main goal of artificial intelligence and machine learning is to make the machines to behave in such a way that if that task was being done by a human, it should have involved the use of intelligence [77].

Genetic programming which is an evolutionary computation (EC) technique, is answering one of the central questions in computer science (raised by Arthur Samuel in 1959), about how to have computers to learn and solve problems without giving them explicit instructions and without explicitly programming them [45]. Genetic programming (GP), is like a genetic algorithm (GA) but the individuals in the population that are being evolved are computer programs. Every individual in the population will be evolved in a random process, similar to Darwinian natural selection, in order to breed a better (and fitter) population of individuals [44]. Although GP results

can not be guaranteed since it uses a stochastic process, it has been very successful in finding novel solutions for many different problems [74]. Prior to applying genetic programming to a problem, terminals and functions set have to be determined. After this, a function for measuring the fitness of individuals should be selected and then a condition/criterion has to be defined in order to determine when to terminate the run. Figure 5.1 shows the general flow of a GP. Genetic Programming generally follows the following steps:

1. Create a random initial population of programs.
2. Execute each program, evaluate it and assign its fitness value.
3. Using a selection method which is based on the fitness values of the individuals, select one or more individuals from the population, in order to breed the new population.
4. Using a reproduction operation, generate the new individuals from the selected ones in the last step.
5. If a solution is found or any terminating condition is met, go to (6); otherwise go to (2).
6. Return the best individual so far.

There are a few terms that are used in the above GP description. These terms are explained below.

### 5.1.1 Function and Terminal Sets

Defining functions and terminals is one of the most important parts of GP design. As the name suggests, functions are the valid functions of the program. In most cases they are simple primitive math operators such as addition and subtraction. Terminals are random constants and independent variables that hold the external inputs. Combining terminals and functions, expression trees are generated. In these trees, functions are operators and terminals are operands. Therefore, each tree can generate a value.

### 5.1.2 Selection Method

As explained above, GP has a number of generations. At the end of each generation, the fitness values of the individuals in the population are calculated. At this step

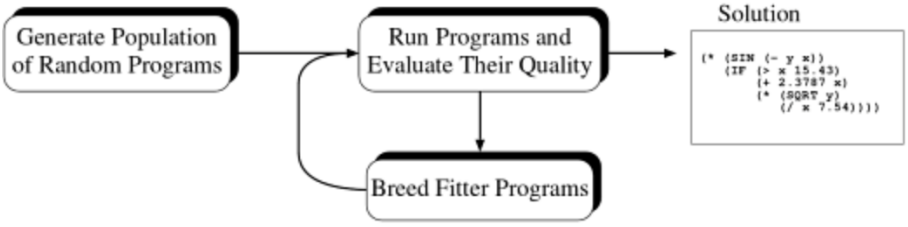


Figure 5.1: A fitness function is used to evaluate the individuals. There are two reproduction operators (Crossover and Mutation) that will breed new population based on the fitness of current individuals. Image from [74].

selection methods are used to select fitter individuals in order to breed new individuals for the next generation. The difference in selection methods is in the way they select individuals for reproduction. In a GP program *elitism* can also be used in which the best individuals will remain in the population of the next generation unchanged (i.e. without being modified by reproduction operators). Two well-known selection methods are *Tournament Selection* and *Roulette-Wheel Selection*.

### 5.1.3 Reproduction Operators

Reproduction operators are in charge of breeding new individuals for the next generation. At the end of each generation individuals will be evaluated using a fitness function. Then, using a selection method, a number of individuals in the current generation will be selected and passed to the reproduction methods to breed the individuals for the next generation. The two reproduction methods are *mutation* and *crossover*. Mutation brings diversity to the population and makes it possible to explore the search space. In mutation, a node in the tree will be selected and the nodes in that sub-tree will change their values to other valid values at random locations. Crossover, which brings the ability to exploit the search space, works on two individuals and produces two children. In crossover an entire sub-tree of the first individual will be replaced with another sub-tree from the second individual. More accurately, two sub-trees are swapped and generate two new individuals.

## 5.2 Databases, Toolkits and Platforms

In this work a variety of information from different databases has been used and analyzed with different toolkits and platforms. The following sections explain these data sources and platforms.

### 5.2.1 Genotator

There is a huge amount of genetic data that is increasing over time. There are numerous data resources that collect and organize different biological datasets. Researchers often have to manually browse and integrate data from various sources to obtain information about a particular disease. Genotator [91] which is developed at Harvard University, aims at collecting and integrating data from well known genetic resources. Based on the information that Genotator collects, it generates reliable gene-to-disease rankings for any disease. In this work, Genotator was used to obtain the set of known disease genes for our diseases of interest.

### 5.2.2 Cytoscape

Cytoscape is a major computational platform that provides the means for analyzing biological networks. Cytoscape is a free open source software and has accessible API (application programming interface) which has made this possible for software developers to develop their own plugins for this platform [76], which brings more functionality to Cytoscape and makes it more flexible. Using these plugins, different types of biological datasets can be obtained, analyzed and visualized [11]. For creating the input datasets of our GP program, two plugins in Cytoscape have been used: GeneMANIA and CentiScaPe.

### 5.2.3 GeneMANIA

GeneMANIA is a gene function prediction tool that analyzes a given list of genes (as the input) and expands the list and prioritizes the genes based on the available proteomic and genomic datasets [92]. The homo sapiens (human) interaction database of GeneMANIA records different types of interactions (co-expression, protein–protein interaction, pathway, co-localization etc.) based on 395 different interaction databases developed to date. It contains the interactions among 21435 genes from the human genome.

GeneMANIA produces a ranking of genes based on generated scores that are higher for genes that share the same functionality or properties with the query genes. This approach is based on the principle of *guilt by association* as described in Section 4.2.1. GeneMANIA is also freely available as a plugin for Cytoscape [58]. In this work, GeneMANIA is used to extract a list of candidate genes that closely interact with the known disease genes. This approach uses different evidence types, available

for the users of GeneMANIA. Hence, it uses a composite network generated by GeneMANIA that combines interaction networks of those evidence types. The genes in this composite network are separated into training and testing datasets to be used by the GP program; see Section 7.1 for further details. The evidence types used in this work are co-expression, co-localization, genetic interactions, pathway, physical interactions, predicted functional relationships and shared protein domains. These evidence types are explained in the following sections based on the explanations provided by GeneMANIA [25].

### Co-expression

Represents the data collected from the gene expression. In this sense, two genes that have similar levels of expression are considered to be connected. Most of the data from gene expression studies are collected from Gene Expression Omnibus (GEO) [20].

### Co-localization

This is mostly the location based data, such as proteins in proximity or genes that are expressed in the same tissue. Based on this, two genes will be connected if they are in a known module or in a same cellular tissue.

### Genetic Interactions

This is the data collected from genetic interactions. This is mainly regarding functionally related genes, which means modifications in one gene, can effect the other gene as well. BioGRID [84] is a repository for interaction datasets that is mainly used by GeneMANIA for collecting interaction data.

### Pathway

Data collected from different pathways in which gene products are linked to each other if they are interacting in the same pathway. According to the definition in [37], a pathway is a series of actions between different molecules that lead to a certain product or a certain change. The pathway data in GeneMANIA are collected from different data sources such as BioCyc and Reactome via PathwayCommons [9].

#### Physical Interactions

Physical interaction is what is known as protein-protein interaction, described earlier in Section 3.1.5. Based on this, genes will be linked if in a protein-protein interaction study they are found to be interacting. GeneMANIA uses different protein interaction databases to collect these data, two of which are BioGRID [84] and PathwayCommons [9].

#### Predicted Functional Relationships

This evidence type is based on predicting relationships between genes, which is often regarding protein interactions. The main source of these predictions is based on the known functional interactions of the genes in other organisms.

#### Shared Protein Domains

Proteins have different domains and families. In these type of data, the gene products that have the same protein data will be linked to each other. GeneMANIA uses various databases for protein domains, such as InterPro [36], SMART [49] and Pfam [48].

### 5.2.4 CentiScaPe

With the increasing amount of experimental data and large amount of available datasets, there is an increasing need for analyzing them. These datasets are frequently presented as huge biological networks, where nodes represent biological entities and the links between them (edges) show interactions between those entities [11]. Analyzing these complex networks can help to extract different information. For example, the network around a known disease gene can be analyzed to find the genes that closely interact with that known disease gene, and based on the aforementioned principle of guilt by association, those genes can be identified as candidate disease genes [86]. Here, a similar idea is used. In order to find the genes that closely interact with the known disease genes, several centrality measures for the nodes in the network surrounding the known disease genes is calculated. Analyzing these centrality measures together can give us different information such as important nodes in that network.

In this work, CentiScaPe is used to compute the centrality measures for the nodes of our composite network. CentiScaPe is a bioinformatics tool that is freely available as a plugin for Cytoscape. It computes different network centralities and inte-

grates them into Cytoscape, so they can be used with other Cytoscape plugins. The computed centrality parameters in CentiScaPe are Degree, Average Distance, Stress, Diameter, Betweenness, Closeness, Radiality, Eccentricity and Centroid Value [80]. These centrality measures are described in Section 4.3.

### 5.2.5 ECJ

We use ECJ to implement genetic programming. ECJ is a Java based evolutionary computation (EC) framework, developed by Sean Luke et al. in 1998 at George Mason University [47]. ECJ was created after massive modifications on lil-gp [96] which showed that it is not possible to further extend that platform. At this time, ECJ is a strong and stable platform with lots of features (like island models, coevolution, multi objective optimization algorithms, etc.) and is widely used for genetic programming [54].

## 5.3 Experiment Description

In our GP approach we analyze the complex network around the already known disease genes and extract several centrality measures for every node in this network. Based on these measures, our GP approach works to find patterns in the known disease genes and then tries to find these patterns in the other genes of this network. Scores are assigned to genes based on the similarities of their patterns, and are used for ranking. The output is a ranking that prioritizes the genes based on the likelihood of their involvement in a given genetic disease.

In machine learning there are two main approaches to learning: supervised and unsupervised. In supervised learning, data is divided into two groups, input and output, and the goal is to predict the value of output based on the input data. In order to achieve this goal, a training set (a set of input data with known output values) will be used to assist the learning process. Two examples of supervised learning are regression and classification. In the former there is a continuous output value, and in the latter, a discrete output value to be predicted. In unsupervised learning, there is no training or testing dataset (no dataset with known output) and the effort is on finding structures or relations in data [23]. One of the examples of unsupervised learning is data clustering in which the goal is to find groups of data for which their members are very similar. Our approach is an example of supervised learning.

Genetic Programing has great potential for classification and has been successfully

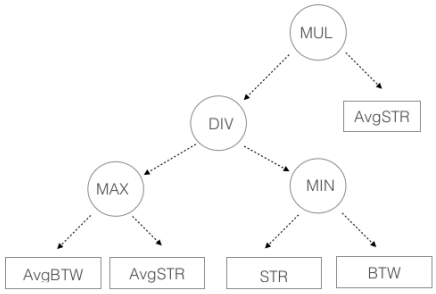


Figure 5.2: A sample expression tree.

applied to many different supervised and unsupervised learning tasks [23, 69, 19]. Classification requires a training dataset, used to induce a classifier, and a testing dataset, used to measure the quality of the classifier.

The input data, obtained from GeneMANIA, must be separated into training and testing datasets (see Section 7.1). It is important that the training dataset is not biased and includes the same number of members from different classes of data.

### 5.3.1 GP Language

Each GP program has a set of terminals and a function set, which it combines to generate the expression trees. Our terminal set contains 9 attributes corresponding to the 9 centrality measures for each of the genes in the aforementioned training/testing datasets. These 9 centrality measures are the ones computed in CentiScaPe, namely degree, average distance, stress, diameter, betweenness, closeness, radiality, eccentricity and centroid value (described in Section 4.3). There are 6 functions in the function set, listed in Table 5.1. These function list covers the most basic math operators and as reported in next two chapters, they were sufficient for GP to perform well on the input dataset.

Based on these, the GP expression tree is combined of the terminals which hold values of centrality measures, and math operators (listed in Table 5.1). Hence, each tree generates a value. These values will be considered as scores for genes and based on them, genes are assigned a rank in the rank list. Figure 5.2 shows a sample expression tree. These expression trees grow over generations and the evolved tree at the final generation is much bigger in size than the depicted sample.

Table 5.1: GP Functions

| Name | Function |
|---|---|
| Add | Addition of two numbers |
| Sub | Subtraction of two numbers |
| Div | Division of two numbers, returns 0 if the 2nd number is 0 |
| Mul | Multiplication of two numbers |
| Min | Minimum of two numbers |
| Max | Maximum of two numbers |

### 5.3.2 Fitness Evaluation

Defining a suitable fitness function and finding a proper way to evaluate the individuals is one the most important aspects of GP application, that has significant impact on the results. Fitness can be calculated in many different ways, such as number of errors, number of hits, total time, etc. In this work the effort is on generating meaningful scores based on the centrality measures. One of the meaningful scores that was available here, was the gene score from GeneMANIA. These are assigned to genes based on their shared functionalities and properties with known disease genes. Therefore, in the evaluation of generated individuals and the scores which they return, effort is taken to see how close they get to the scores from GeneMANIA. An ideal solution here: (i) finds the pattern in the centrality measures of the nodes in our composite network, (ii) predicts the important nodes of this complex network; and (iii) reports them as candidate disease genes.

Koza fitness was used in this program. This fitness holds the fitness of individuals, as described in Koza I [44]. Standardized fitness ranges from 0 (best fitness) to infinity (worst possible) and adjusted fitness is computed based on that:

$$f_{adj} = \frac{1}{1 + f_{stn}} \tag{5.1}$$

where $f_{adj}$ is the adjusted fitness and $f_{stn}$ is the standard fitness. Although Koza fitness prints out the adjusted fitness, it stores the standardized fitness, where lower values are considered as better and fitter individuals.

As explained in previous section (Section 5.3.1), a rank list is created for the genes. Rank of each gene in this rank list is compared with its origial rank in GeneMANIA's rank list. If the distance is less than 10, it will be considered as a hit, otherwise the distance will be added to a sum of error parameter. This sum of error will be passed to the fitness function as the standard fitness and then adjusted fitness will be calculated based on that.

## 5.4 Benchmark Tests

### 5.4.1 Leave-One-Out Cross Validation

In order to test the abilities of our method, we tested it using the leave-one-out cross validation test. This is a standard and widely used procedure that tests the given method against already known disease genes.

If we name the set of known disease genes $K$, we run this procedure $|K|$ times, each time for a different known disease gene in $K$. If we have a set of candidate disease genes $C$, at each round of this procedure, a gene $g$ will be removed from $K$ and will be added to $C$. Next, the method will run normally on $C \cup g$ and a ranking will be generated for these genes. The goal is to see if the gene $g$ will be ranked as a highly associative gene (ranked lower than a predefined threshold)[50].
Considering the difficulty of evaluating the generated rankings of different computational methods for candidate disease genes, leave-one-out gives a simple and effective approach to test these methods with known disease genes. Because of this, leave-one-out is used elsewhere e.g. [94], [90] and [42].

### 5.4.2 Fold Enrichment Analysis

Following the leave-one-out cross-validation test, fold enrichment analysis is usually used to report the results. Based on the provided explanations in [94], this analysis simply says that if the method under study ranks n% of the known disease genes in the top m% of the candidate genes (the aforementioned threshold), there would be a $n/m$-fold enrichment on average.

As required by this analysis, a threshold must be defined to separate the two prediction classes (highly associative genes and peripheral genes). There is no specific rule to determine the threshold and as it is evident, it would be harder for the methods under study to perform well on smaller thresholds. It should be noted that for comparison of several methods, the same threshold must be used in these tests.

### 5.4.3 Receiver-Operating Characteristic (ROC) Analysis

*Receiver-operating characteristic* or ROC analysis is another way to analyze the obtained results via leave-one-out cross-validation test. This analysis basically measures the performance of a binary classifier. In this analysis we define a threshold $T$ for our generated ranking for candidate disease genes (see Section 5.4.2). This threshold on a generated rank list specifies the two prediction classes. Based on this, considering

sets $K$ and $C$ and known disease gene $g$ as defined in Section 5.4.1, if $g$ is ranked lower than $T$, it will be considered as a successfully identified known disease gene and we call it a *True-Positive* or TP. If $g$ is ranked higher than $T$, the method is failed at identifying this known disease gene and we call it a *False-Negative* or FN. Based on the computed values of TP and FN, the *sensitivity* of our computational method will be defined as equation 5.2:

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.2}$$

The sensitivity value simply reports the success rate of our method (percentage of identified genes) on the set of known disease genes.

# Chapter 6

# Case Study: Breast Cancer

In this chapter a case study of Breast Cancer based on our approach is presented. Breast Cancer is the most common type of cancer among women which accounts for 22% of their cancers [94]. In the United States it accounts for more than 40000 deaths per year [3] and in the United Kingdom it has been reported to be the most common cause of death among women aged 40 to 50 [57].

Breast cancer is a genetic disease which is caused by mutation and alteration in genes or their expression [41]. It is currently estimated that about 25% of the risk of breast cancer is explained by the known susceptibility genes for this disease. Moreover these genes explain less than 5% of breast cancer incidence [68], suggesting a need for more studies on the genetic basis of this disease. Our method is applied to the problem of finding Breast Cancer genes and in this chapter the experiments are explained.

## 6.1 Input Data

Using OMIM [56], 15 susceptibility genes were selected to be involved in breast cancer and they are used as the known disease genes in these experiments. Table 6.1 outlines these known disease genes. These genes are the same as those used in [85] and [94]. The only exception was that these other works used an extra gene in addition to these 15 known disease genes. This additional gene could not be identified by GeneMANIA which was the main reason that it was not used here.

The GeneMANIA plugin for Cytoscape was used to extract the top 2000 genes interacting around these known disease genes and the composite network in which they interact. This composite network was generated by GeneMANIA using the following evidence types: co-expression, co-localization, genetic interactions, pathway,

Table 6.1: Known Disease Genes for Breast Cancer

| Gene | NCBI ID |
|---|---|
| BRCA1 | 672 |
| AR | 367 |
| ATM | 472 |
| CHEK2 | 11200 |
| BRCA2 | 675 |
| STK11 | 6794 |
| RAD51 | 5888 |
| PTEN | 5728 |
| BARD1 | 580 |
| TP53 | 7157 |
| RB1CC1 | 9821 |
| NCOA3 | 8202 |
| PIK3CA | 5290 |
| PPM1D | 8493 |
| CASP8 | 841 |

physical interactions (protein-protein interactions), predicted functional relationships and shared protein domains. These 2000 genes and the set of our known disease genes is used for training and testing of our GP program.

## 6.2 GP Design

The GP method has 4 separate working sets, each containing 4 islands that work together. The purpose is to aid in dividing the computational effort and making the runs as fast as possible. Also, as will be explained later, in some experiments each set is used to implement different settings and use different centrality measures as input data, to compare the numerous possible setups of this problem. The experiments test the following:

1. GP parameters settings
2. Fitness evaluation
3. Different sets of centrality measures to be analyzed for each node

One of the main goals is to improve the performance of the GP in the leave-one-out cross validation test.

The first experiment uses the following centrality measures: eccentricity, radiality, closeness, degree and their averages. As explained in Section 4.3, these centrality measures, when they are accompanied by their averages, are expected to generate meaningful results. Table 6.2 shows the GP parameters of this experiment and Table 6.3 shows the migration pattern of the island models. Islands were set to send individuals every 4 generations.

Table 6.2: GP Parameters of First Experiment

| **Parameter** | **Value** |
|---|---|
| number of generations | 50 |
| number of jobs | 20 (5 per set) |
| number of threads | 8 |
| population size | 2500 (per island) |
| selection | Tournament (Size = 7) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 1014 |
| size of testing dataset | 1001 |
| number of elite individuals | 10 |

Table 6.3: Island Model Migration Pattern

| **From** | **To** | **No. of individuals** |
|---|---|---|
| Island 1 | Island 2 | 20 |
| Island 2 | Island 3 | 20 |
| Island 3 | Island 4, 1 | 50 |
| Island 4 | Island 1 | 25 |

In this experiment, each set had 5 jobs, but since everything (except the random values) was the same for each set, we had a total of 20 runs. Figure 6.1 shows the averages of mean fitness and best fitness values of the GP program for 20 runs, over 50 generations. In this experiment, GP was trying to generate scores for the genes, based on the generated scores by GeneMANIA, and then rank each gene based on the generated scores. The main problem in here was that the generated scores by GeneMANIA were not normally distributed and the difference of their scores, for the last 1500 genes (out of 2015 genes in total) was less than 1, while for the top genes there was a difference of 100 for each gene. Based on this distribution GP was trying to generate a score to cover those last 1500 genes, ignoring the top high-scoring genes. As a result, the generated fitness values were better than they should be. This brought us to the next experiment.

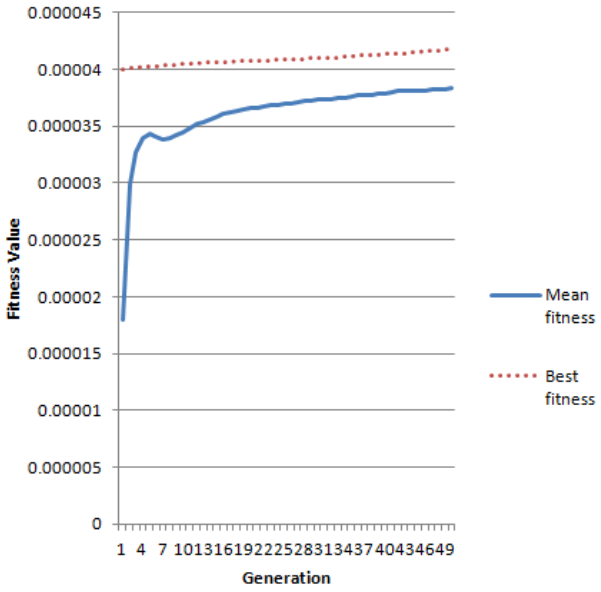


Figure 6.1: Average of the mean fitness and best fitness of 20 runs over 50 generations. Breast cancer, experiment 1.

## 6.3 Using GeneMANIA Ordering of the Genes

The next experiment was an effort to value the order of the genes in the GeneMANIA results instead of their scores. Based on this, in this experiment GP is trying to generate an ordering of the genes based on their order in the GeneMANIA ranklist of the top 2000 genes for breast cancer. If the distance between the generated order of a gene with its original order in GeneMANIA list is less than 10, GP will count it as a hit. Having this in mind, the second experiment uses the same parameters settings as in Tables 6.2 and 6.3 with the new evaluation and scoring system. Figure 6.2 depicts the convergence of mean fitness and best fitness of 20 jobs, over 50 generations.

In order to compare the fitness values of these two first experiments, an unpaired two sample t-test assuming unequal variances was used. The null hypothesis was that the means of the two experiments were equal and the P-value was 0.05. Table 6.4 shows the result of this test. Based on this, the great value of t-Stat shows a huge difference in the two groups. Also, if we check this value with the 't Critical two-tail' it is greater than this, which rejects our null hypothesis. By checking the mean values, it can be seen that the first experiments had better fitness values. Please note that these reported values are adjusted fitness values which should be in this range: $0 < f_{adj} \leq 1$ with 0 being the worst and 1 the best fitness values.

Although in this new scoring system the fitness values are worse, we still kept it

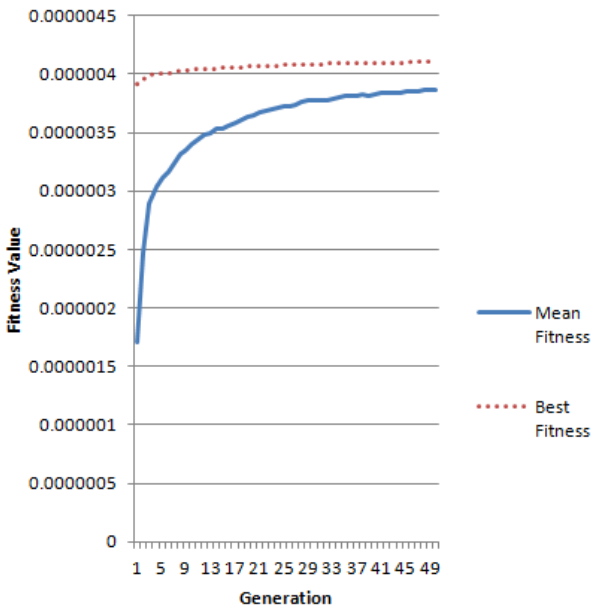


Figure 6.2: Average of the mean fitness and best fitness of 20 runs over 50 generations. Breast cancer, experiment 2.

for the next experiments, since the results and the generated fitness values are more realistic in this case.

Table 6.4: t-test assuming unequal variances over 20 runs of the first two experiments of breast cancer. P-value was set to 0.05

| | **Experiment 1** | **Experiment 2** |
|---|---|---|
| Mean | 4.20708E-05 | 4.11731E-06 |
| Variance | 1.25524E-12 | 1.88669E-16 |
| Observations | 20 | 20 |
| Hypothesized Mean Difference | 0 | |
| df | 19 | |
| t Stat | 151.4854957 | |
| P(T<=t) one-tail | 4.71714E-31 | |
| t Critical one-tail | 1.729132792 | |
| P(T<=t) two-tail | 9.43428E-31 | |
| t Critical two-tail | 2.09302405 | |

## 6.4 Different Centrality Measures

As explained earlier in Section 4.3, different centrality measures can be more meaningful when they are accompanied with some other specific centrality measures. In

the next experiment, different sets used different combinations of centrality measures, in order to compare their output and decide about the best centrality measures to use. Table 6.5 outlines the centrality measures used by each set in the third experiment.

Table 6.5: Centrality Measures Used in Third Experiment of Breast Cancer

| **Set No.** | **Centrality Measures** |
|---|---|
| Set 1 | Eccentricity, Radiality, Closeness, Degree and their averages |
| Set 2 | All Centrality Measures and their averages |
| Set 3 | Stress, Betweenness and their averages |
| Set 4 | Degree, Eigen Vector and their averages |

Table 6.6 shows the GP parameters of this experiment. The migration pattern of the island model is same as for the first experiment, outlined in Table 6.3.

Table 6.6: GP Parameters of Third Experiment, Breast Cancer

| **Parameter** | **Value** |
|---|---|
| number of generations | 100 |
| number of jobs | 5 |
| number of threads | 2 |
| population size | 2500 (per island) |
| selection | Tournament (Size = 5) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 1014 |
| size of testing dataset | 1001 |
| number of elite individuals | 10 |

In order to get the results of the leave-one-out cross validation test, this experiment was executed 15 times. Each time one of the known disease genes was given to the testing set and the other 14 genes were given to the training set, making a testing set of 1001 genes and a training set of 1014 genes. Each time the GP program generated a ranking for the testing genes and the objective was to see if that single known disease gene in the testing set can rank high.

As mentioned in Table 6.6, the number of jobs is 5 and each run of this experiment generates 5 rankings (for each set). After running this experiment 15 times, we had 75 rankings for these 15 known disease genes for each set. The rank of each known disease gene in these 75 rankings was extracted and compared with those of the other sets to compare the performance of each set. It should be noted that in these rankings lower is better and ideally all these 15 known disease genes should be ranked $1^{st}$ in their own separate run.

Table 6.7: t-test assuming unequal variances to compare different sets of experiment 3 of breast cancer. P-value was 0.05 for all the tests.

| **t-test No. 1** | **Set 1** | **Set 2** |
|---|---|---|
| Mean | 125.94 | 286.9 |
| Variance | 22941.3376 | 94626.46 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 108 | |
| t Stat | -4.0654105 | |
| P(T<=t) one-tail | 4.5651E-05 | |
| t Critical one-tail | 1.65908514 | |
| P(T<=t) two-tail | 9.1302E-05 | |
| t Critical two-tail | 1.98217342 | |

| **t-test No. 2** | **Set 1** | **Set 3** |
|---|---|---|
| Mean | 125.94 | 69.56 |
| Variance | 22941.33757 | 5714.628 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 109 | |
| t Stat | 2.884351465 | |
| P(T<=t) one-tail | 0.002363794 | |
| t Critical one-tail | 1.658953459 | |
| P(T<=t) two-tail | 0.004727589 | |
| t Critical two-tail | 1.98196743 | |

| **t-test No. 3** | **Set 1** | **Set 4** |
|---|---|---|
| Mean | 125.94 | 116.1133 |
| Variance | 22941.3376 | 22795.86 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 148 | |
| t Stat | 0.3979258 | |
| P(T<=t) one-tail | 0.34562914 | |
| t Critical one-tail | 1.65521451 | |
| P(T<=t) two-tail | 0.69125827 | |
| t Critical two-tail | 1.97612246 | |

| **t-test No. 4** | **Set 3** | **Set 4** |
|---|---|---|
| Mean | 69.56 | 116.1133 |
| Variance | 5714.628108 | 22795.86 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 109 | |
| t Stat | -2.38769651 | |
| P(T<=t) one-tail | 0.009337583 | |
| t Critical one-tail | 1.658953459 | |
| P(T<=t) two-tail | 0.018675166 | |
| t Critical two-tail | 1.98196743 | |

We used 4 different t-tests (unpaired two sample t-test, assuming unequal variances) to compare 'set 1 and set 2', 'set 1 and set 3', 'set 1 and set 4', and 'set 3 and set 4'. Table 6.7 shows the results of these 4 t-tests. Based on these results, set 1 which is using eccentricity, radiality, closeness and degree performed better than set 2 which uses all 9 centrality measures listed in Section 5.2.4. One reason can be that giving all of these centrality measures to the GP is overwhelming it with too much information so that it is unable to generate good results compared to an experiment in which we give it a selected number of centrality measures. Furthermore, the tests show that set 3, using stress and betweenness, outperforms sets 1 and 4 with a noticeable difference in the mean values of its generated rankings, compared to sets 1 and 4. Also the t-test on sets 1 and 4 shows that their performance is almost similar and the differences in their mean values are negligible. Therefore set 3, using stress and betweenness, generated the best ranks and set 2, using all centrality measures, was the worst. Set 1 using radiality, eccentricity, closeness, degree and set 4 using degree and eigenvector had the same performance.

It should be mentioned that in these series of 15 runs for the leave-one-out cross validation test, the threshold for the known disease gene to be considered as identified in our GP ranking was set to 25 (top 2.5%, as in [85]) and 4 out of 15 known disease genes were successfully identified in this experiment.

## 6.5 Improving the Results

Based on the previous experiments, we implemented the last experiment to get the best possible results. As we see in the previous section (6.4), the best combination of centrality measures was stress, betweenness and their averages. For this experiment we used these centrality measures for all the sets. Table 6.8 outlines the parameter settings of the last experiment and Table 6.9 shows the migration patterns of islands which is the same for all four sets. The mail-box capacity was set to 200.

Table 6.8: GP Parameters of Last/Best Experiment, Breast Cancer

| **Parameter** | **Value** |
|---|---|
| number of generations | 100 |
| number of jobs | 20 (5 per set) |
| number of threads | 4 |
| population size | 1024 (256 per island) |
| selection | Tournament (Size = 3) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 214 |
| size of testing dataset | 1801 |
| number of elite individuals | 5 |

Table 6.9: Island Model Migration Pattern

| **From** | **To** | **No. of individuals** |
|---|---|---|
| Island 1 | Island 2, 3, 4 | 20 |
| Island 2 | Island 1, 3, 4 | 20 |
| Island 3 | Island 1, 2, 4 | 20 |
| Island 4 | Island 1, 2, 3 | 20 |

Figure 6.3 shows the convergence of average mean fitness and best fitness for this experiment over 100 generations, for all 20 runs of this experiment. Another important change in this experiment was the number of training and testing sets. The reason for this change was to make sure that the size of the training set is not too large for GP, so it can perform well. Hence, out of 2015 genes for breast cancer, 214 were given to the training set and the rest were given to the testing set.

In order to check the effect of smaller training size on the fitness values, we used a two sample t-test on this experiment and the previous one (experiment 3). Table 6.10 shows the result of this t-test. As we can see, the t-Stat shows a huge difference in the mean fitness values of the best runs of these experiments. As we see in this

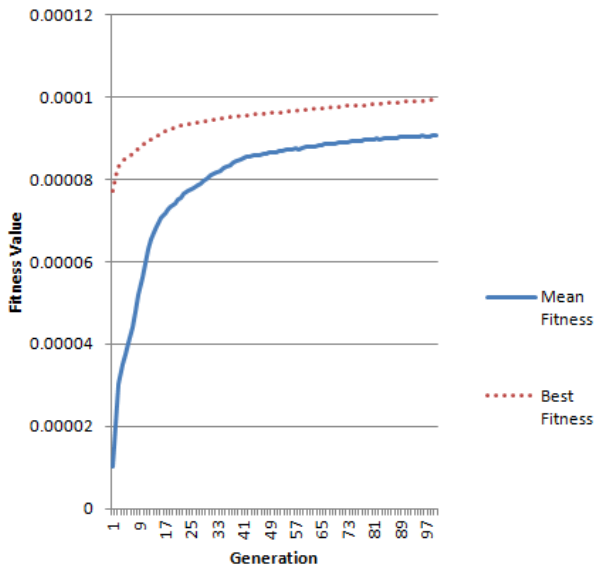


Figure 6.3: Average of the mean fitness and best fitness of 20 runs over 100 generations. Breast cancer, final experiment

table, the mean value of experiment 4 is about 24 times greater than the previous one which shows a huge improvement in the fitness values.

Table 6.10: Two sample t-test assuming unequal variances, comparing third and fourth experiments of breast cancer.

| | **Experiment 3** | **Experiment 4** |
|---|---|---|
| Mean | 4.139838E-06 | 9.95502E-05 |
| Variance | 2.75799E-15 | 1.66538E-12 |
| Observations | 20 | 20 |
| Hypothesized Mean Difference | 0 | |
| df | 19 | |
| t Stat | -330.3670629 | |
| P(T<=t) one-tail | 1.74726E-37 | |
| t Critical one-tail | 1.729132792 | |
| P(T<=t) two-tail | 3.49453E-37 | |
| t Critical two-tail | 2.09302405 | |

### 6.5.1 Leave-One-Out Cross Validation

As mentioned earlier in Section 5.4.1, in order to run this test we have to run our GP program 15 times and each time we give one of the known disease genes to the testing set and the other 14 known disease genes to the training set. Each time our

GP generates a ranking for the genes in the testing set and if that known disease gene is ranked lower than a certain threshold, we count it as a successfully identified gene. There is no specific rule to set this threshold, but in order to be able to make a fair comparison with other work, we set the threshold to be the top 2.5% of the genes, which is the same threshold that is used in [85] and in [94].

Our method, using stress and betweenness as the centrality measures for each gene, successfully predicted 9 out of 15 known disease genes. These genes are listed in Table 6.11.

Table 6.11: Successfully Predicted Genes via GP from the 15 Known Disease Genes for Breast Cancer listed in Table 6.1

| Gene | Median Ranking | Best Ranking | Standard Deviation |
|---|---|---|---|
| TP53 | 4 | 2.5 | 4.98 |
| BARD1 | 5 | 3 | 19.39 |
| BRCA1 | 6.75 | 6 | 22.42 |
| PIK3CA | 9.75 | 7 | 8.81 |
| ATM | 15.75 | 14 | 9.08 |
| NCOA3 | 24 | 19 | 13.11 |
| RAD51 | 34 | 29 | 36.30 |
| CASP8 | 42 | 32 | 21.75 |
| RB1CC1 | 45.25 | 42 | 15.84 |

### 6.5.2 Comparison with Other Works

We compared these results with the earlier work performed in [85] and also with CIPHER [94], on the same set of candidate disease genes and same set of known disease genes. Note that as mentioned in Section 6.1, both these previous works had an extra known disease gene in addition to those of this experiment. Table 6.12 shows the results of the benchmark tests.

The GA approach has the best performance here, identifying 12 out of 16 known disease genes. Our GP approach has the same performance as CIPHER with the same percentage of identified known disease genes (same sensitivity values).

### 6.5.3 Predicting novel disease genes for breast cancer

After running the benchmark tests on our method and comparing it with other work, we analyzed the generated ranklists for the candidate genes. Some of the genes that

Table 6.12: Analysis and Comparison to Other Frameworks

| **Framework** | **Fold enrich.** | **Leave-one-out** | **Sensitivity** |
|---|---|---|---|
| CIPHER | 25 | 10/16 | 0.625 |
| GA approach | 30 | 12/16 | 0.75 |
| GP approach | 24 | 9/15 | 0.60 |

were consistently ranked high in most of ranklists, found to be mentioned in different works to be involved in breast cancer. It shows that these genes and also other highly ranked genes are worthy of being further studied.

One of these genes is APP (Amyloid-$\beta$ precursor protein) that in a recent study ([53]) the pathological role of this gene in breast cancer has been studied. This gene has been reported to be related for different types of cancer, but this recent study showed its role in pathogenesis of breast cancer. There is also CYP19A1 (cytochrome P450) and XRCC3 ( X-ray repair complementing defective repair in Chinese hamster cells 3) that they are both reported by genetics home reference [67] that changes in these genes are associated with breast cancer.

SNURF (SNRPN upstream reading frame) is another highly ranked gene in most of our rankings that is studied in [79] and [75]. It is mentioned that this gene cooperates with other genes and can cause a breast cancer. Another highly ranked protein coding gene is TSPY1 (testis specific protein, Y-linked 1). This gene is reported in different studies such as [28] and [22] to be related to breast cancer. There are also two genes ESR1 (estrogen receptor 1) and EP300 (E1A binding protein p300) that both were ranked high in earlier work done in [85] and they are reported to have some supporting evidence of involvement in breast cancer, based on different data resource according to Genotator.

# Chapter 7

# Case Study: Parkinson's Disease

The methodology was applied to the problem of finding genes involved in Parkinson's Disease, a neurological condition that in most cases is due to an interaction between many genes and environmental risk factors [81]. Due to the complicated nature of these interactions, it is thus a particularly worthwhile but challenging disease to study for disease-gene association.

## 7.1 Input Data

Using Genotator, we extracted 15 genes known to be involved in Parkinson's disease. They are the same set of known disease genes that are used in [85] and they are listed in Table 7.1. These are high-scoring genes scored by Genotator based on their involvement in Parkinson's disease.

Using the GeneMANIA plugin for Cytoscape, a list of the 3000 top-scoring genes that closely interact with the 15 known disease genes for Parkinson's disease, and the composite network in which they interact, was extracted. The composite network from GeneMANIA is based on different evidence types that are available for users of this software, namely: co-expression, co-localization, genetic interactions, pathway, physical interactions (protein-protein interactions), predicted functional relationships and shared protein domains. Using CentiScaPe, the nine centrality measures were generated for each node of this network.

The 3015 genes (15 known disease genes from Genotator, and 3000 genes from GeneMANIA) were divided into two groups, one for training and the other for testing. If the genes in each dataset were to have similar scores (for example, if all genes with high scores were in one dataset and all genes with lower scores in the other), then there would be a high probability that GP would produce biased results. Having this

Table 7.1: Known Disease Genes

| Gene | NCBI ID |
|---|---|
| LRRK2 | 120892 |
| SNCA | 6622 |
| PARK2 | 5071 |
| MAPT | 4137 |
| APOE | 348 |
| GBA | 2629 |
| GAK | 2580 |
| BST1 | 683 |
| DRD2 | 1813 |
| PINK1 | 65018 |
| MAOB | 4129 |
| BDNF | 627 |
| CYP2D6 | 1565 |
| PON1 | 5444 |
| COMT | 1312 |

in mind, the genes for the training and testing datasets were selected so that both datasets have genes with high scores and low scores. Note that the sizes of training and testing datasets vary in different experiments and they will be mentioned for each experiment.

## 7.2 GP Design

Similar to the design of our GP method for the previous case study (see Section 6.2), our GP has 4 separate working sets, each containing four asynchronous islands. This special implementation drastically reduced the running time of the experiments, making us able to run more different experiments. Also because of having different separate working sets, it was possible to use different approaches in each set for research purposes.

For the first experiment, we used the parameters listed in Table 7.2. The islands of all four sets use the migration pattern mentioned in Table 7.3, and based on this pattern, every four generations they send their individuals to other islands. For this experiment, we used all the available centrality measures listed in Section 5.2.4, to see the power of them all together for this problem. Note that in this experiment everything except the random values remained the same for each set.

Table 7.2: GP Parameters of First Experiment, Parkinson's Disease

| **Parameter** | **Value** |
|---|---|
| number of generations | 51 |
| number of jobs | 20 (5 per set) |
| number of threads | 8 |
| population size | 2500 (per island) |
| selection | Tournament (Size = 7) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 1514 |
| size of testing dataset | 1501 |
| number of elite individuals | 10 |

Table 7.3: Island Model Migration Pattern

| **From** | **To** | **No. of individuals** |
|---|---|---|
| Island 1 | Island 2 | 20 |
| Island 2 | Island 3 | 20 |
| Island 3 | Island 4, 1 | 50 |
| Island 4 | Island 1 | 25 |

Figure 7.1 illustrates the convergence of average mean fitness and best fitness for this experiment over 51 generations. In order to see the performance of this experiment on the genes, we performed the leave-one-out cross validation test on our set of 15 known disease genes. The threshold was set to the top 1.8%, the same threshold used to report and compare other methods in Section 7.12. Out of the 15 known disease genes, this method identified two of them (SNCA and APOE) and showed a potential for further improvement.

Unexpectedly, using all available centrality measures did not result in an outstanding performance in the leave-one-out cross validation test. The concern here is to check whether changing centrality measures can actually help with these results or not. Having this in mind, the next experiment is an attempt to see the effect of using a different set of centrality measures in analyzing our composite network of genes' interactions. Based on this, for the second experiment everything was the same as the first one (see Tables 7.2 and 7.3), but this time a new set of centrality measures was used: radiality, eccentricity, closeness, degree and their averages. Based on the studies on the centrality measures and explanations in Section 4.3, and also based on the results of the Breast Cancer case study, these centrality measures were expected to work better together.

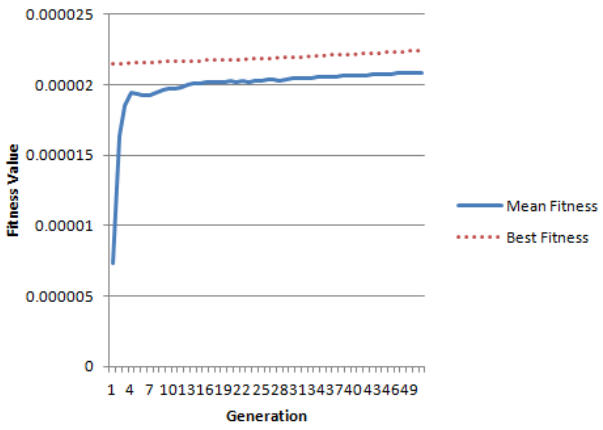


Figure 7.1: Average of the mean fitness and best fitness of 20 runs over 51 generations. Parkinson's disease, experiment 1.

After running this experiment 15 times, each time for a known disease gene of our known disease genes set, this method successfully identified 4 out of 15 known disease genes. In order to have a comprehensive comparison of this experiment and the previous one, we used a two sample t-test on all 15 known disease genes, in which lower is better. As Table 7.4 shows, the t-test shows that there is a difference between the two experiments in favour of the second experiment. Based on this result, it can be inferred that the second experiment not only outperformed the first experiment on the four identified known disease genes, but it performed better on the rest of the known disease genes as well. Considering that the second experiment was identical to the first one, except for the set of centrality measures that was used, this proves that using a selected set of centrality measures can actually have a great effect on the final results of our method.

## 7.3 Different Sets of Centrality Measures

The results of the second experiment showed the importance of having the right set of centrality measures to analyze the composite network of genes interactions. The next experiment was designed to compare the performance of different sets of centrality measures on identifying known disease genes. Based on the explanations in Section 4.3 and based on the results of the Breast Cancer case study, we used different sets of centrality measures for our third experiment. The centrality measures used by each set are outlined in Table 7.5.

This experiment was performed using a slightly different set of parameter settings

Table 7.4: Unpaired two sample t-test assuming unequal variances, comparing first and second experiments of Parkinson's disease. P-value is 0.05.

| | **Experiment 1** | **Experiment 2** |
|---|---|---|
| Mean | 519.6733333 | 228.9 |
| Variance | 231272.6182 | 94352.54054 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 126 | |
| t Stat | 4.412921791 | |
| P(T<=t) one-tail | 1.08316E-05 | |
| t Critical one-tail | 1.657036982 | |
| P(T<=t) two-tail | 2.16631E-05 | |
| t Critical two-tail | 1.978970576 | |

Table 7.5: Centrality Measures Used in Third Experiment of Parkinson's Disease

| **Set No.** | **Centrality Measures** |
|---|---|
| Set 1 | Eccentricity, Radiality, Closeness, Degree and their averages |
| Set 2 | Stress, Betweenness and their averages |
| Set 3 | Stress, Betweenness and their averages |
| Set 4 | Degree, Eigen-Vector and their averages |

and also used a different migration pattern for the island models, as reported in Tables 7.6 and Table 7.7 respectively.

After running the experiment, an unpaired two sample t-test was used to compare the performance of each set, using different sets of centrality measures. The comparison was based on each of the 15 known disease genes, in the leave-one-out cross validation test to see which set produced lower ranks for these known disease genes. Table 7.8 shows the results of these t-tests. As we see, and as was expected, sets 2 and 3 had the same results with a small non-significant difference. Set 1 and set 4 also had the same performance and the difference in their mean values was non-significant. Moreover, set 2, which used stress and betweenness, showed a better performance compared to set 1, which used radiality, eccentricity, closeness and degree. Finally, the result of the t-test comparing sets 2 and 4 suggests that there is no significant difference between these two (although the 't-stat' is very close to the critical value). However, if we consider the results of one-tail, the difference is significant in favour of set 2. Based on these results, it can be inferred that the combination of stress, betweenness and their averages, outperformed other combinations of centrality measures. These combination of centrality measures were expected to work well together (see Section 4.3) and they also worked well for our Breast Cancer

Table 7.6: GP Parameters of Third Experiment, Parkinson's Disease

| Parameter | Value |
|---|---|
| number of generations | 100 |
| number of jobs | 5 |
| number of threads | 4 |
| population size | 3000 (per island) |
| selection | Tournament (Size = 3) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 1514 |
| size of testing dataset | 1501 |
| number of elite individuals | 10 |

Table 7.7: Island Model Migration Pattern

| From | To | No. of individuals |
|---|---|---|
| Island 1 | Island 2, 3, 4 | 20 |
| Island 2 | Island 1, 3, 4 | 20 |
| Island 3 | Island 1, 2, 4 | 20 |
| Island 4 | Island 1, 2, 3 | 20 |

case study.

## 7.4 Improving the Results

The final experiment was designed and implemented based on the results of previous experiments to obtain the best possible output of this method. As we experienced in previous sections, centrality measures play an important role in the success of our method. Based on the result of the previous experiment (Section 7.3), stress, betweenness and their averages were used by all sets. There were also a few changes in parameter settings. The important change in here was the reduced size of the training set, as it was expected to improve the fitness values in our experiment. Table 7.9 shows these parameters for the last experiment. Islands used the same pattern as for the previous example (Table 7.7) to send individuals to each other. Figure 7.2 shows the convergence of the average mean fitness and best fitness for this experiment over 100 generations, for all 20 runs of this experiment.

The fitness values of this experiment were compared with the previous one using a t-test. Table 7.10 shows the result of this t-test. Based on this, there was a huge improvement in the fitness values of the fourth experiment. This shows the effect of

Table 7.8: t-test assuming unequal variances to compare different sets of experiment 3 of Parkinson's disease. P-value is 0.05 for all of the tests.

| **t-test No. 1** | **Set 2** | **Set 3** |
|---|---|---|
| Mean | 174.4933333 | 168.8866667 |
| Variance | 102591.7128 | 102641.7268 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 148 | |
| t Stat | 0.107179396 | |
| P(T<=t) one-tail | 0.457395906 | |
| t Critical one-tail | 1.655214507 | |
| P(T<=t) two-tail | 0.914791813 | |
| t Critical two-tail | 1.976459531 | |

| **t-test No. 2** | **Set 2** | **Set 1** |
|---|---|---|
| Mean | 174.4933333 | 319.98 |
| Variance | 102591.7128 | 113245.753 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 148 | |
| t Stat | -2.712004518 | |
| P(T<=t) one-tail | 0.003739975 | |
| t Critical one-tail | 1.655214507 | |
| P(T<=t) two-tail | 0.007479949 | |
| t Critical two-tail | 1.976122461 | |

| **t-test No. 3** | **Set 2** | **Set 4** |
|---|---|---|
| Mean | 174.4933333 | 276.5466667 |
| Variance | 102591.7128 | 133995.4877 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 145 | |
| t Stat | -1.81703044 | |
| P(T<=t) one-tail | 0.35638687 | |
| t Critical one-tail | 1.655430252 | |
| P(T<=t) two-tail | 0.071277374 | |
| t Critical two-tail | 1.97645931 | |

| **t-test No. 4** | **Set 1** | **Set 4** |
|---|---|---|
| Mean | 319.98 | 276.5466667 |
| Variance | 113245.753 | 133995.4877 |
| Observations | 75 | 75 |
| Hypothesized Mean Difference | 0 | |
| df | 147 | |
| t Stat | 0.756472832 | |
| P(T<=t) one-tail | 0.225288203 | |
| t Critical one-tail | 1.655285437 | |
| P(T<=t) two-tail | 0.450576405 | |
| t Critical two-tail | 1.976233277 | |

choosing a smaller training set.

### 7.4.1 Leave-One-Out Cross Validation Test

For this test, we performed our method 15 separate times. Each time one of the known disease genes was given to the testing set and the other 14 were given to the training set. Each time, the GP method produced a ranking for the genes in training set. If that known disease gene in the testing set was ranked lower than a certain threshold, it was counted as a successfully identified known disease gene. The threshold in this

Table 7.9: GP Parameters of Last Experiment, Parkinson's Disease

| **Parameter** | **Value** |
|---|---|
| number of generations | 100 |
| number of jobs | 20 |
| number of threads | 4 |
| population size | 1024 (256 per island) |
| selection | Tournament (Size = 3) |
| cross-over rate | 90% |
| mutation rate | 10% |
| size of training dataset | 314 |
| size of testing dataset | 2701 |
| number of elite individuals | 5 |

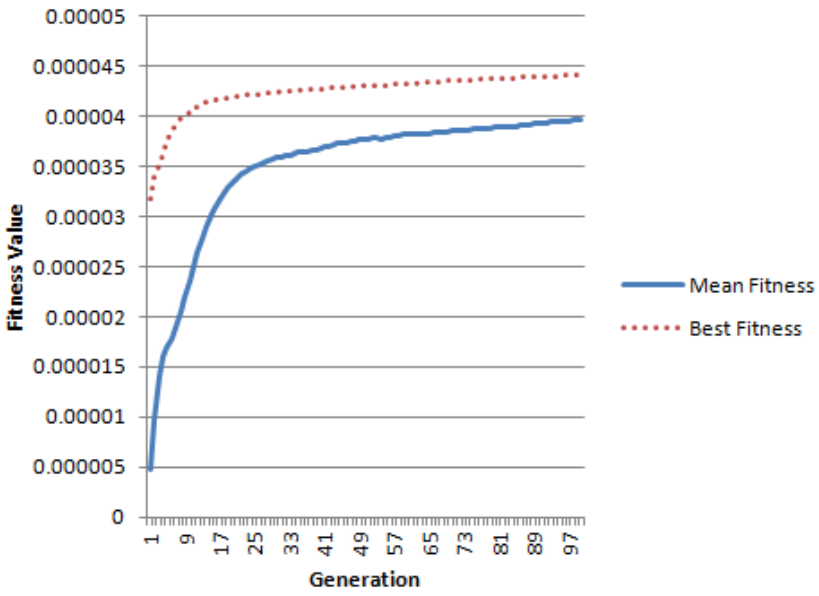


Figure 7.2: Average of the mean fitness and best fitness of 20 runs over 100 generations. Parkinson's disease, last experiment.

experiment was set to the top 50 genes (the same threshold that is used to report and compare other methodologies in Section 7.12).

Our method using stress, betweenness and their averages successfully identified 6 out of 15 known disease genes. These genes and their ranks are reported in Table 7.11.

### 7.4.2 Comparison with Other Works

We compared the obtained results with the earlier work performed in [85] and also with ENDEAVOUR [2] on the same set of known disease genes. Table 7.12 shows the result of this comparison. Based on these results, our GP approach outperformed the other two approaches, identifying 6 out of 15 known disease genes. It should be noted that as expected, our approach has different performance on different diseases. In the breast cancer case study, GA performed better than the GP approach but for the Parkinson's disease, GP outperformed the GA approach.

### 7.4.3 Predicting novel Parkinson's disease genes

After leave-one-out cross-validation on the known disease genes, we analyzed the ranklist generated by our GP program to look at its prediction of novel disease genes.

Table 7.10: An unpaired two sample t-test, assuming unequal variances comparing fitness values of third and fourth experiments.

| | **Experiment 3** | **Experiment 4** |
|---|---|---|
| Mean | 1.81952E-06 | 4.42633E-05 |
| Variance | 8.09145E-17 | 1.05503E-13 |
| Observations | 20 | 20 |
| Hypothesized Mean Difference | 0 | |
| df | 19 | |
| t Stat | -584.158556 | |
| P(T<=t) one-tail | 3.46483E-42 | |
| t Critical one-tail | 1.729132792 | |
| P(T<=t) two-tail | 6.92966E-42 | |
| t Critical two-tail | 2.09302405 | |

Table 7.11: Successfully Predicted Genes via GP from the 15 Known Disease Genes for Parkinson's Disease listed in Table 7.1

| **Gene** | **Median Ranking** | **Best Ranking** | **Standard Deviation** |
|---|---|---|---|
| PARK2 | 5 | 4.5 | 21.24 |
| PINK1 | 5.75 | 4.5 | 184.00 |
| BST1 | 5.75 | 5 | 54.72 |
| SNCA | 20.75 | 8 | 263.60 |
| PON1 | 49 | 35 | 181.33 |
| MAOB | 50.5 | 34.5 | 110.70 |

Interestingly, some of the top ranked genes of our GP program are mentioned in different works and studies to be relevant to Parkinson's disease.

### Epidermal Growth Factor Receptor (EGFR)

The epidermal growth factor receptor was one of the highest ranked genes in all of the runs of our program. In studying the literature, we found that Chen-Plotkin et al. [10] and Pellecchia et al. [72] have observed this gene to be related to some cognitive inabilities of Parkinson's disease patients. Putting all of this evidence together, it makes EGFR gene a good candidate disease gene for Parkinson's disease that should be further studied.

Table 7.12: Analysis and Comparison to Other Frameworks

| Framework | Fold enrich. | Leave-one-out | Sensitivity |
|---|---|---|---|
| ENDEAVOUR | 11.11 | 3/15 | 0.2 |
| GA approach | 18.33 | 5/15 | 0.33 |
| GP approach | 22.22 | 6/15 | 0.4 |

### Complement Component 8, alpha polypeptide (C8A)

C8A is a component of the complement system which encodes the alpha subunit of C8, and mutations in this gene can lead to complement C8 alpha-gamma deficiency [61]. This gene is ranked among the top genes in different runs of our GP method and is reported in different works to be related to Parkinson's disease. In PDbase, a database for Parkinson's Disease-related genes, it is mentioned that this gene is in the Parkinson's disease pathway, along with 8 other genes [71]. BioGraph [52] which integrates heterogeneous knowledge bases, has ranked C8A $4^{th}$ out of 18180 genes (top 0.02%) in the context of Parkinson's disease pathway [5].

### Growth Associated Protein 43 (GAP-43)

GAP-43 is a protein associated with nerve growth, and is generated by the GAP-43 gene in humans (NCBI gene ID: 2596) [62, 82]. There are some studies about this gene and its protein being involved in Parkinson's disease (in both humans and animals). Seth et al. in [83] have studied the molecular events, triggered by 6-Hydroxydopamine (6-OHDA) exposure and the expression of growth associated protein GAP-43. 6-OHDA has been implicated in the pathogenesis of Parkinson's disease [40]. Interestingly, GAP-43 is one of the highly ranked genes in different runs of our GP method, which makes it a good candidate gene for Parkinson's disease to be further investigated.

# Chapter 8

# Conclusion and Future Work

The increasing amount of data in biological datasets brings with it the requirement for analyzing the data to extract useful information. These datasets are often presented in the form of complex networks. Due to the complexity of these networks, classic algorithms for graphs will no longer work in a reasonable time, therefore other approaches should be used.

In this work, a GP approach is used to analyze the complex network around the known disease genes. Using different sets of centrality measures, GP tries to extract useful information, and based on them it generates a ranking for the genes. A lower rank for a gene means that it has a higher probability of being involved in the disease under study.

We presented two case studies for breast cancer and Parkinson's disease. We performed our method on a set of known disease genes for breast cancer and Parkinson's disease and then we used two well known and reliable frameworks to compare the results. The results of both experiments showed the great potential of GP in processing complex networks and for the problem of disease-gene association. As outlined in Tables 6.12 and 7.12, our method compares favourably to results obtained by CIPHER, ENDEAVOUR and also the earlier work done with GA in [85].

As it was expected for the GP to find patterns in known disease genes and use them to find candidate disease genes, our method was successful in generating acceptable and meaningful results. As an example, although this method sometimes fails to identify some known disease genes, for the well known disease genes such as SNCA and PARK2 for Parkinson's disease or TP53, BRCA1 and ATM for Breast Cancer it always performs well, as it consistently ranks them among the top 10 genes for all of the runs. Another example is the effect of different centrality measures on the outcome of our method. As elaborated earlier in Section 4.3, some centrality

measures are expected to work better and generate more meaningful results when they are combined. Based on the experiments explained in our case studies (Chapters 6 and 7), as we expected, centrality measures had significant impact on the results. For both genetic diseases that we studied (breast cancer and Parkinson's disease), the best combination of centrality measures were stress, betweenness and their averages. It should be noted that these centrality measures may not be the best combination for other genetic diseases and also there are still some other possible combinations to test out of the 9 available centrality measures. Furthermore, there are some other powerful centrality measures that are not available in CentiScaPe. Analyzing gene interaction networks with those centrality measures may help GP in making better decisions for candidate disease genes.

Although the results are satisfactory, there is still a need for improvement. GP parameters were determined empirically in this work, and it is possible to find better parameter settings for the GP program. Also the GP language can be modified and enhanced as it may make a huge difference in GP performance. Fitness evaluation is another important part of every GP program that can have a significant effect on the results. Different criteria should be considered and studied in order to find more powerful fitness functions.

As mentioned before, this GP method is based on the island models. The architecture of these islands and their migration patterns and parameter settings are other important factors that can be improved. This work is generic and disease centred, which means that it can be applied to almost any genetic disease with some modifications. Hence, applying this method to more genetic diseases can be considered in the future, especially since we expect different performance and accuracy for this method on different genetic diseases.

# Bibliography


[1] Abigail Acland, Richa Agarwala, Tanya Barrett, Jeff Beck, Dennis A Benson, Colleen Bollin, Evan Bolton, Stephen H Bryant, Kathi Canese, Deanna M Church, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 42(D1):D7, 2014.

[2] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Van Loo, Bert Coessens, Frederik De Smet, Leon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, et al. Gene prioritization through genomic data fusion. *Nature biotechnology*, 24(5):537–544, 2006.

[3] Muhammad Al-Hajj, Max S Wicha, Adalberto Benito-Hernandez, Sean J Morrison, and Michael F Clarke. Prospective identification of tumorigenic breast cancer cells. *Proceedings of the National Academy of Sciences*, 100(7):3983–3988, 2003.

[4] Joanna Amberger, Carol A Bocchini, Alan F Scott, and Ada Hamosh. Mckusick's online mendelian inheritance in man (omim®). *Nucleic acids research*, 37(suppl 1):D793–D796, 2009.

[5] Universiteit Antwerpen and VIB. Biograph: Discovering biomedical relations by unsupervised hypothesis generation. `http://staging.biograph.be/concept/graph/C1521736/C1413023`, 2015. [Online; accessed 15-April-2015].

[6] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[7] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

[8] Sarah Calvo, Mohit Jain, Xiaohui Xie, Sunil A Sheth, Betty Chang, Olga A Goldberger, Antonella Spinazzola, Massimo Zeviani, Steven A Carr, and Vamsi K Mootha. Systematic identification of human mitochondrial disease genes through integrative genomics. *Nature genetics*, 38(5):576–582, 2006.

[9] Memorial Sloan-Kettering Cancer Center and the University of Toronto. Pathway commons. `http://www.pathwaycommons.org/about/`, 2015. [Online; accessed 14-July-2015].

[10] Alice S Chen-Plotkin, William T Hu, Andrew Siderowf, Daniel Weintraub, Rachel Goldmann Gross, Howard I Hurtig, Sharon X Xie, Steven E Arnold, Murray Grossman, Christopher M Clark, et al. Plasma epidermal growth factor levels predict cognitive decline in parkinson's disease. *Annals of neurology*, 69(4):655–663, 2011.

[11] Melissa S Cline, Michael Smoot, Ethan Cerami, Allan Kuchinsky, Nerius Landys, Chris Workman, Rowan Christmas, Iliana Avila-Campilo, Michael Creech, Benjamin Gross, et al. Integration of biological networks and gene expression data using cytoscape. *Nature protocols*, 2(10):2366–2382, 2007.

[12] Wikimedia Commons. Dna structure and bases color. `http://commons.wikimedia.org/wiki/File:DNA_structure_and_bases_color_FR.svg`, 2015. [Online; accessed 14-July-2015].

[13] Wikimedia Commons. Protein structures. `http://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Main_protein_structure_levels_en.svg/434px-Main_protein_structure_levels_en.svg.png`, 2015. [Online; accessed 15-April-2014].

[14] Wikimedia Commons. Rna internal loop. `https://en.wikipedia.org/wiki/File:C-loop-secondary-structure.svg`, 2015. [Online; accessed 15-September-2015].

[15] Wikimedia Commons. Rna stem-loop structure. `https://en.wikipedia.org/wiki/File:Stem-loop.svg`, 2015. [Online; accessed 15-September-2015].

[16] Wikimedia Commons. Rna structure. `http://commons.wikimedia.org/wiki/File:RNA-comparedto-DNA_thymineAndUracilCorrected.png`, 2015. [Online; accessed 20-April-2015].

[17] Francis Crick et al. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[18] Cytoscape. Cytoscape: An open source platform for complex network analysis and visualization. `http://www.cytoscape.org/images/top_slides/humanInteractome1_700px.png`, 2015. [Online; accessed 20-April-2015].

[19] Ivanoe De Falco, Ernesto Tarantino, Antonio Della Cioppa, and Francesco Fontanella. An innovative approach to genetic programmingbased clustering. In *Applied Soft Computing Technologies: The Challenge of Complexity*, pages 55–64. Springer, 2006.

[20] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

[21] ENDEAVOUR. Endeavour web server. `http://homes.esat.kuleuven.be/~bioiuser/endeavour/tool/endeavourweb.php`, 2015. [Online; accessed 15-September-2015].

[22] Mirjam T Epping, Lars AT Meijer, Oscar Krijgsman, Johannes L Bos, Pier Paolo Pandolfi, and René Bernards. Tspyl5 suppresses p53 levels and function by physical interaction with usp7. *Nature Cell Biology*, 13(1):102–108, 2011.

[23] Pedro G Espejo, Sebastián Ventura, and Francisco Herrera. A survey on the application of genetic programming to classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(2):121–144, 2010.

[24] Ernesto Estrada and Örjan Bodin. Using network centrality measures to manage landscape connectivity. *Ecological Applications*, 18(7):1810–1825, 2008.

[25] Donnelly Centre for Cellular and Biomolecular Research. Explanations of evidence types. `http://pages.genemania.org/help/`, 2015. [Online; accessed 14-July-2015].

[26] Simon J Furney, Borja Calvo, Pedro Larranaga, Jose A Lozano, and Nuria Lopez-Bigas. Prioritization of candidate cancer genesan aid to oncogenomic studies. *Nucleic Acids Research*, 36(18):e115–e115, 2008.

[27] Kyle J Gaulton, Karen L Mohlke, and Todd J Vision. A computational system to select candidate genes for complex human traits. *Bioinformatics*, 23(9):1132–1140, 2007.

[28] Peter Geck. Symptotic detection of chimerism: Y does it matter? *Chimerism*, 4(4):144–146, 2013.

[29] Cosmas Giallourakis, Charlotte Henson, Michael Reich, Xiaohui Xie, and Vamsi K Mootha. Disease gene discovery through integrative genomics. *Annu. Rev. Genomics Hum. Genet.*, 6:381–406, 2005.

[30] Jesse Gillis and Paul Pavlidis. guilt by association is the exception rather than the rule in gene networks. *PLoS computational biology*, 8(3):e1002444, 2012.

[31] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[32] Shuba Gopal, Anne Haake, Rhys Price Jones, and Paul Tymann. *Bioinformatics: a computing perspective*. McGraw-Hill Science/Engineering/Math, 2008.

[33] Shapiro Group. Rna secondary structure prediction with pseudoknots. `http://cylofold.abcc.ncifcrf.gov`, 2015. [Online; accessed 15-September-2015].

[34] Ada Hamosh, Alan F Scott, Joanna S Amberger, Carol A Bocchini, and Victor A McKusick. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(suppl 1):D514–D517, 2005.

[35] Trey Ideker and Roded Sharan. Protein networks in disease. *Genome Research*, 18(4):644–652, 2008.

[36] European Bioinformatics Institute. Protein sequence, analysis and classification. `http://www.ebi.ac.uk/interpro/`, 2015. [Online; accessed 15-June-2015].

[37] National Human Genome Research Institute. athway definition. `https://www.genome.gov/27530687`, 2015. [Online; accessed 14-July-2015].

[38] University of Vienna Institute for Theoretical Chemistry. Prediction of secondary structures of single stranded rna or dna sequences. `http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi`, 2015. [Online; accessed 15-September-2015].

[39] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, et al. Kegg for linking genomes to life and the environment. *Nucleic acids research*, 36(suppl 1):D480–D484, 2008.

[40] Farouk Karoum, Stanislaw J Chrapusta, Michael F Egan, and Richard Jed Wyatt. Absence of 6-hydroxydopamine in the rat brain after treatment with stimulants and other dopaminergic agents: A mass fragmentographic study. *Journal of neurochemistry*, 61(4):1369–1375, 1993.

[41] RS Khane. Genetic basis of breast cancer. *Journal of clinical and diagnostic research*, 6:3–498, 2012.

[42] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N Robinson. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 82(4):949–958, 2008.

[43] Sebastian Köhler, Sandra C Doelken, Christopher J Mungall, Sebastian Bauer, Helen V Firth, Isabelle Bailleul-Forestier, Graeme CM Black, Danielle L Brown, Michael Brudno, Jennifer Campbell, et al. The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic acids research*, page gkt1026, 2013.

[44] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[45] John R Koza. Survey of genetic algorithms and genetic programming. In *Wescon Conference Record*, pages 589–594. WESTERN PERIODICALS COMPANY, 1995.

[46] The Isambert Lab. Rna/dna folding predictions including pseudoknots and entangled helices. `http://kinefold.curie.fr`, 2015. [Online; accessed 15-Seotember-2015].

[47] ECLab Evolutionary Computation Laboratory. Ecj website. `http://cs.gmu.edu/~eclab/projects/ecj/`, 2015. [Online; accessed 14-March-2015].

[48] European Molecular Biology Laboratory. A large collection of protein families. `http://pfam.xfam.org`, 2015. [Online; accessed 14-July-2015].

[49] European Molecular Biology Laboratory. Simple modular architecture research tool. `http://smart.embl-heidelberg.de`, 2015. [Online; accessed 14-July-2015].

[50] Duc-Hau Le and Yung-Keun Kwon. Gpec: a cytoscape plug-in for random walk-based gene prioritization and biomedical evidence collection. *Computational biology and chemistry*, 37:17–23, 2012.

[51] Heng Li and Nils Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473–483, 2010.

[52] Anthony M Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, Jurgen Del-Favero, et al. Biograph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biol*, 12(6):R57, 2011.

[53] Seunghwan Lim, Byoung Kwon Yoo, Hae-Suk Kim, Hannah L Gilmore, Yonghun Lee, Hyun-pil Lee, Seong-Jin Kim, John Letterio, and Hyoung-gon Lee. Amyloid-beta precursor protein promotes cell proliferation and motility of advanced breast cancer. *BMC cancer*, 14(1):928, 2014.

[54] S Luke. The ecj owners manual–a user manual for the ecj evolutionary computation library. *August*, 2014.

[55] Marco Masseroli, Osvaldo Galati, and Francesco Pinciroli. Gfinder: genetic disease and phenotype location statistical analysis and mining of dynamically annotated gene lists. *Nucleic Acids Research*, 33(suppl 2):W717–W723, 2005.

[56] Victor A McKusick. Mendelian inheritance in man and its online version, omim. *The American Journal of Human Genetics*, 80(4):588–604, 2007.

[57] Klim McPherson, CaMa Steel, and JM Dixon. Abc of breast diseases: breast cancerepidemiology, risk factors, and genetics. *BMJ: British Medical Journal*, 321(7261):624, 2000.

[58] Jason Montojo, Khalid Zuberi, Harold Rodriguez, Farzana Kazi, George Wright, Sylva L Donaldson, Quaid Morris, and Gary D Bader. Genemania cytoscape plugin: fast gene function predictions on the desktop. *Bioinformatics*, 26(22):2927–2928, 2010.

[59] National Institute of Health National Human Genome Research Institute. Genetics home reference. `http://www.genome.gov/20019523`, 2015. [Online; accessed 15-August-2014].

[60] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

[61] NCBI. National centre for biotechnology information. `http://www.ncbi.nlm.nih.gov/gene/731`, 2015. [Online; accessed 20-April-2015].

[62] NCBI. National centre for biotechnology information. `http://www.ncbi.nlm.nih.gov/gene/2596`, 2015. [Online; accessed 20-April-2015].

[63] NovoCIB. Nucleotide metabolism explorer. `http://www.novocib.com/Nucleotides_Life_Science.html`, 2015. [Online; accessed 20-April-2015].

[64] U.S. National Library of Medicine. Genetics home reference. `http://ghr.nlm.nih.gov/handbook/basics/dna`, 2015. [Online; accessed 15-April-2014].

[65] U.S. National Library of Medicine. Genetics home reference. `http://ghr.nlm.nih.gov/glossary=rna`, 2015. [Online; accessed 15-April-2014].

[66] U.S. National Library of Medicine. Genetics home reference. `http://ghr.nlm.nih.gov/glossary=linkageanalysis`, 2015. [Online; accessed 15-August-2014].

[67] U.S. National Library of Medicine. Genetics home reference. `http://ghr.nlm.nih.gov/condition/breast-cancer/show/Related+Gene(s)`, 2015. [Online; accessed 20-June-2015].

[68] RA Oldenburg, Hanne Meijers-Heijboer, CJ Cornelisse, and Peter Devilee. Genetic susceptibility for breast cancer: how many more genes to be found? *Critical reviews in oncology/hematology*, 63(2):125–149, 2007.

[69] Mihai Oltean and Laura Dioşan. An autonomous gp-based system for regression and classification problems. *Applied Soft Computing*, 9(1):49–60, 2009.

[70] Martin Oti and Han G Brunner. The modular nature of genetic diseases. *Clinical genetics*, 71(1):1–11, 2007.

[71] PDbase. Pdbase, a database of parkinson's disease-related genes and genetic variation using substantia nigra ests. `http://bioportal.kobic.re.kr/PDbase/pathway.jsp?pathway=h\_alternativePathway`, 2015. [Online; accessed 20-April-2015].

[72] Maria Teresa Pellecchia, Gabriella Santangelo, Marina Picillo, Rosario Pivonello, Katia Longo, Claudia Pivonello, Carmine Vitale, Marianna Amboni, Anna De Rosa, Marcello Moccia, et al. Serum epidermal growth factor predicts cognitive functions in early, drug-naive parkinsons disease patients. *Journal of neurology*, 260(2):438–444, 2013.

[73] Rosario M Piro and Ferdinando Di Cunto. Computational approaches to disease-gene prediction: rationale, classification and successes. *FEBS Journal*, 279(5):678–696, 2012.

[74] Riccardo Poli, William B Langdon, Nicholas F McPhee, and John R Koza. *A field guide to genetic programming*. Lulu. com, 2008.

[75] Hetti Poukka, Piia Aarnisalo, Henrikki Santti, Olli A Jänne, and Jorma J Palvimo. Coregulator small nuclear ring finger protein (snurf) enhances sp1- and steroid receptor-mediated transcription by different mechanisms. *Journal of Biological Chemistry*, 275(1):571–579, 2000.

[76] Rintaro Saito, Michael E Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, Samad Lotia, Alexander R Pico, Gary D Bader, and Trey Ideker. A travel guide to cytoscape plugins. *Nature methods*, 9(11):1069–1076, 2012.

[77] Arthur L Samuel. Ai, where it has been and where it is going. In *IJCAI*, pages 1152–1157, 1983.

[78] Frederick Sanger, Steven Nicklen, and Alan R Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

[79] Bradley Saville, Hetti Poukka, Mark Wormke, Olli A Jänne, Jorma J Palvimo, Matthew Stoner, Ismael Samudio, and Stephen Safe. Cooperative coactivation of estrogen receptor $\alpha$ in zr-75 human breast cancer cells by snurf and tata-binding protein. *Journal of Biological Chemistry*, 277(4):2485–2497, 2002.

[80] Giovanni Scardoni, Michele Petterlini, and Carlo Laudanna. Analyzing biological network parameters with centiscape. *Bioinformatics*, 25(21):2857–2859, 2009.

[81] Claudia Schulte and Thomas Gasser. Genetic basis of parkinsons disease: inheritance, penetrance, and expression. *The application of clinical genetics*, 4:67, 2011.

[82] LifeMap Sciences. Genecards, human gene database. `http://www.genecards.org/cgi-bin/carddisp.pl?gene=GAP43`, 2015. [Online; accessed 20-April-2015].

[83] K Seth, AK Agrawal, MH Aziz, A Ahmad, Y Shukla, N Mathur, and PK Seth. Induced expression of early response genes/oxidative injury in rat pheochromocytoma (pc12) cell line by 6-hydroxydopamine: implication for parkinson's disease. *Neuroscience letters*, 330(1):89–93, 2002.

[84] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl 1):D535–D539, 2006.

[85] Koosha Tahmasebipour. Disease-gene association using a genetic algorithm. Master's thesis, Faculty of Mathematics and Science, Brock University, August 2014.

[86] Koosha Tahmasebipour and Sheridan Houghten. Disease-gene association using a genetic algorithm. In *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on BioInformatics and BioEngineering*, pages 191–197. IEEE, 2014.

[87] Nicki Tiffin, Miguel A Andrade-Navarro, and Carolina Perez-Iratxeta. Linking genes to diseases: its all in the data. *Genome Med*, 1(8):77, 2009.

[88] MA Van Driel, K Cuelenaere, PPCW Kemmeren, Jack AM Leunissen, Han G Brunner, and Gert Vriend. Geneseeker: extraction and integration of human disease-related information from web-based genetic databases. *Nucleic Acids Research*, 33(suppl 2):W758–W761, 2005.

[89] Marc A van Driel, Jorn Bruggeman, Gert Vriend, Han G Brunner, and Jack AM Leunissen. A text-mining analysis of the human phenome. *European journal of human genetics*, 14(5):535–542, 2006.

[90] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS computational biology*, 6(1):e1000641, 2010.

[91] Dennis P Wall, Rimma Pivovarov, Mark Tong, Jae-Yoon Jung, Vincent A Fusaro, Todd F DeLuca, and Peter J Tonellato. Genotator: a disease-agnostic tool for genetic annotation of disease. *BMC medical genomics*, 3(1):50, 2010.

[92] David Warde-Farley, Sylva L Donaldson, Ovi Comes, Khalid Zuberi, Rashad Badrawi, Pauline Chao, Max Franz, Chris Grouios, Farzana Kazi, Christian Tannus Lopes, et al. The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl 2):W214–W220, 2010.

[93] David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 35(suppl 1):D5–D12, 2007.

[94] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Molecular systems biology*, 4(1), 2008.

[95] Medical Xpress. Constitution of chromosome. `http://medicalxpress.com/news/2015-04-gene-maternal-aneuploidy.html`, 2015. [Online; accessed 20-April-2015].

[96] Douglas Zongker and Bill Punch. lil-gp 1.0 users manual. *Dept. of Computer Science, Michigan State University*, 1, 1995.

# Appendix A

# Leave-One-Out Cross-Validation Results

As mentioned before in Section 5.4, leave-one-out cross validation test was used to test the performance of our computational method. For both case studies (Parkinson's disease and Breast Cancer) this test was performed 15 times for each of the known disease genes. The results of the final (and the best) experiment are reported in here for both diseases.

Note that for each known disease gene, there are five paragraphs representing each of the five jobs. In each paragraph, ranking of that specific gene is extracted for each "set–island". Inside the parentheses job number, set number and island number is reported respectively followed by the known disease gene name and its rank in the generated ranklist for candidate disease genes. Also median of generated ranks by all four islands in each set, is reported in a bold text, right after the ranks of four islands in each set.

## A.1 Breast Cancer

### A.1.1 AR

**Job # 0**

(Job.0.S1.I1) AR - 80; (Job.0.S1.I2) AR - 83; (Job.0.S1.I3) AR - 80; (Job.0.S1.I4) AR - 82 : **81.0** (Job.0.S2.I1) AR - 77; (Job.0.S2.I2) AR - 75; (Job.0.S2.I3) AR - 76; (Job.0.S2.I4) AR - 75 : **75.5** (Job.0.S3.I1) AR - 58; (Job.0.S3.I2) AR - 48; (Job.0.S3.I3) AR - 45; (Job.0.S3.I4) AR - 48 : **48.0** (Job.0.S4.I1) AR - 82; (Job.0.S4.I2) AR - 83; (Job.0.S4.I3) AR - 85; (Job.0.S4.I4) AR - 82 : **82.5**

**Job # 1**

(Job.1.S1.I1) AR - 67; (Job.1.S1.I2) AR - 67; (Job.1.S1.I3) AR - 67; (Job.1.S1.I4) AR - 67 : **67.0** (Job.1.S2.I1) AR - 209; (Job.1.S2.I2) AR - 220; (Job.1.S2.I3) AR - 1314; (Job.1.S2.I4) AR - 220 : **220.0** (Job.1.S3.I1) AR - 107; (Job.1.S3.I2) AR - 104; (Job.1.S3.I3) AR - 102; (Job.1.S3.I4) AR - 157 : **105.5** (Job.1.S4.I1) AR - 86; (Job.1.S4.I2) AR - 86; (Job.1.S4.I3) AR - 152; (Job.1.S4.I4) AR - 86 : **86.0**

**Job # 2**

(Job.2.S1.I1) AR - 94; (Job.2.S1.I2) AR - 95; (Job.2.S1.I3) AR - 94; (Job.2.S1.I4) AR - 95 : **94.5** (Job.2.S2.I1) AR - 53; (Job.2.S2.I2) AR - 48; (Job.2.S2.I3) AR - 53; (Job.2.S2.I4) AR - 55 : **53.0** (Job.2.S3.I1) AR - 68; (Job.2.S3.I2) AR - 70; (Job.2.S3.I3) AR - 65; (Job.2.S3.I4) AR - 68 : **68.0** (Job.2.S4.I1) AR - 56; (Job.2.S4.I2) AR - 86; (Job.2.S4.I3) AR - 82; (Job.2.S4.I4) AR - 43 : **69.0**

**Job # 3**

(Job.3.S1.I1) AR - 100; (Job.3.S1.I2) AR - 100; (Job.3.S1.I3) AR - 101; (Job.3.S1.I4) AR - 93 : **100.0** (Job.3.S2.I1) AR - 86; (Job.3.S2.I2) AR - 84; (Job.3.S2.I3) AR - 87; (Job.3.S2.I4) AR - 87 : **86.5** (Job.3.S3.I1) AR - 88; (Job.3.S3.I2) AR - 90; (Job.3.S3.I3) AR - 87; (Job.3.S3.I4) AR - 95 : **89.0** (Job.3.S4.I1) AR - 87; (Job.3.S4.I2) AR - 84; (Job.3.S4.I3) AR - 84; (Job.3.S4.I4) AR - 87 : **85.5**

**Job # 4**

(Job.4.S1.I1) AR - 71; (Job.4.S1.I2) AR - 779; (Job.4.S1.I3) AR - 783; (Job.4.S1.I4) AR - 780 : **779.5** (Job.4.S2.I1) AR - 88; (Job.4.S2.I2) AR - 88; (Job.4.S2.I3) AR - 91; (Job.4.S2.I4) AR - 86 : **88.0** (Job.4.S3.I1) AR - 68; (Job.4.S3.I2) AR - 83; (Job.4.S3.I3) AR - 69; (Job.4.S3.I4) AR - 71 : **70.0** (Job.4.S4.I1) AR - 109; (Job.4.S4.I2) AR - 110; (Job.4.S4.I3) AR - 115; (Job.4.S4.I4) AR - 82 : **109.5**

*Best ranks of each set–island*: 80 - 75 - 45 - 82 - 67 - 209 - 102 - 86 - 94 - 48 - 65 - 43 - 93 - 84 - 87 - 84 - 71 - 86 - 68 - 82 -
*Median*: 85.75
*Median of best*: 82.0

### A.1.2 ATM

**Job # 0**

(Job.0.S1.I1) ATM - 26; (Job.0.S1.I2) ATM - 28; (Job.0.S1.I3) ATM - 26; (Job.0.S1.I4) ATM - 26 : **26.0** (Job.0.S2.I1) ATM - 22; (Job.0.S2.I2) ATM - 22; (Job.0.S2.I3) ATM - 22; (Job.0.S2.I4) ATM - 22 : **22.0** (Job.0.S3.I1) ATM - 5; (Job.0.S3.I2) ATM - 2; (Job.0.S3.I3) ATM - 5; (Job.0.S3.I4) ATM - 2 : **3.5** (Job.0.S4.I1) ATM - 7; (Job.0.S4.I2) ATM - 7; (Job.0.S4.I3) ATM - 9; (Job.0.S4.I4) ATM - 8 : **7.5**

**Job # 1**

(Job.1.S1.I1) ATM - 18; (Job.1.S1.I2) ATM - 17; (Job.1.S1.I3) ATM - 17; (Job.1.S1.I4) ATM - 17 : **17.0** (Job.1.S2.I1) ATM - 21; (Job.1.S2.I2) ATM - 16; (Job.1.S2.I3) ATM - 14; (Job.1.S2.I4) ATM - 17 : **16.5** (Job.1.S3.I1) ATM - 10; (Job.1.S3.I2) ATM - 10; (Job.1.S3.I3) ATM - 10; (Job.1.S3.I4) ATM - 7 : **10.0** (Job.1.S4.I1) ATM - 6; (Job.1.S4.I2) ATM - 6; (Job.1.S4.I3) ATM - 6; (Job.1.S4.I4) ATM - 6 : **6.0**

**Job # 2**

(Job.2.S1.I1) ATM - 15; (Job.2.S1.I2) ATM - 14; (Job.2.S1.I3) ATM - 14; (Job.2.S1.I4) ATM - 14 : **14.0** (Job.2.S2.I1) ATM - 7; (Job.2.S2.I2) ATM - 8; (Job.2.S2.I3) ATM - 7; (Job.2.S2.I4) ATM - 8 : **7.5** (Job.2.S3.I1) ATM - 27; (Job.2.S3.I2) ATM - 45; (Job.2.S3.I3) ATM - 27; (Job.2.S3.I4) ATM - 25 : **27.0** (Job.2.S4.I1) ATM - 13; (Job.2.S4.I2) ATM - 13; (Job.2.S4.I3) ATM - 13; (Job.2.S4.I4) ATM - 13 : **13.0**

**Job # 3**

(Job.3.S1.I1) ATM - 19; (Job.3.S1.I2) ATM - 7; (Job.3.S1.I3) ATM - 11; (Job.3.S1.I4) ATM - 11 : **11.0** (Job.3.S2.I1) ATM - 18; (Job.3.S2.I2) ATM - 18; (Job.3.S2.I3) ATM - 18; (Job.3.S2.I4) ATM - 18 : **18.0** (Job.3.S3.I1) ATM - 5; (Job.3.S3.I2) ATM - 6; (Job.3.S3.I3) ATM - 5; (Job.3.S3.I4) ATM - 5 : **5.0** (Job.3.S4.I1) ATM - 5; (Job.3.S4.I2) ATM - 24; (Job.3.S4.I3) ATM - 23; (Job.3.S4.I4) ATM - 22 : **22.5**

**Job # 4**

(Job.4.S1.I1) ATM - 19; (Job.4.S1.I2) ATM - 26; (Job.4.S1.I3) ATM - 20; (Job.4.S1.I4) ATM - 29 : **23.0** (Job.4.S2.I1) ATM - 16; (Job.4.S2.I2) ATM - 15; (Job.4.S2.I3) ATM - 14; (Job.4.S2.I4) ATM - 15 : **15.0** (Job.4.S3.I1) ATM - 37; (Job.4.S3.I2) ATM - 35; (Job.4.S3.I3) ATM - 36; (Job.4.S3.I4) ATM - 37 : **36.5** (Job.4.S4.I1) ATM - 38; (Job.4.S4.I2) ATM - 27; (Job.4.S4.I3) ATM - 24; (Job.4.S4.I4) ATM - 38 : **32.5**

*Best ranks of each set–island*: 26 - 22 - 2 - 7 - 17 - 14 - 7 - 6 - 14 - 7 - 25 - 13 - 7 - 18 - 5 - 5 - 19 - 14 - 35 - 24 -
*Median*: 15.75
*Median of best*: 14.0

### A.1.3 BARD1

**Job # 0**
(Job.0.S1.I1) BARD1 - 3; (Job.0.S1.I2) BARD1 - 11; (Job.0.S1.I3) BARD1 - 6; (Job.0.S1.I4) BARD1 - 4 : **5.0** (Job.0.S2.I1) BARD1 - 7; (Job.0.S2.I2) BARD1 - 6; (Job.0.S2.I3) BARD1 - 3; (Job.0.S2.I4) BARD1 - 4 : **5.0** (Job.0.S3.I1) BARD1 - 4; (Job.0.S3.I2) BARD1 - 2; (Job.0.S3.I3) BARD1 - 2; (Job.0.S3.I4) BARD1 - 4 : **3.0** (Job.0.S4.I1) BARD1 - 10; (Job.0.S4.I2) BARD1 - 10; (Job.0.S4.I3) BARD1 - 10; (Job.0.S4.I4) BARD1 - 12 : **10.0**

**Job # 1**
(Job.1.S1.I1) BARD1 - 5; (Job.1.S1.I2) BARD1 - 10; (Job.1.S1.I3) BARD1 - 10; (Job.1.S1.I4) BARD1 - 5 : **7.5** (Job.1.S2.I1) BARD1 - 3; (Job.1.S2.I2) BARD1 - 3; (Job.1.S2.I3) BARD1 - 3; (Job.1.S2.I4) BARD1 - 3 : **3.0** (Job.1.S3.I1) BARD1 - 4; (Job.1.S3.I2) BARD1 - 4; (Job.1.S3.I3) BARD1 - 3; (Job.1.S3.I4) BARD1 - 4 : **4.0** (Job.1.S4.I1) BARD1 - 5; (Job.1.S4.I2) BARD1 - 5; (Job.1.S4.I3) BARD1 - 5; (Job.1.S4.I4) BARD1 - 5 : **5.0**

**Job # 2**
(Job.2.S1.I1) BARD1 - 4; (Job.2.S1.I2) BARD1 - 3; (Job.2.S1.I3) BARD1 - 2; (Job.2.S1.I4) BARD1 - 2 : **2.5** (Job.2.S2.I1) BARD1 - 5; (Job.2.S2.I2) BARD1 - 5; (Job.2.S2.I3) BARD1 - 5; (Job.2.S2.I4) BARD1 - 5 : **5.0** (Job.2.S3.I1) BARD1 - 7; (Job.2.S3.I2) BARD1 - 10; (Job.2.S3.I3) BARD1 - 4; (Job.2.S3.I4) BARD1 - 7 : **7.0** (Job.2.S4.I1) BARD1 - 3; (Job.2.S4.I2) BARD1 - 3; (Job.2.S4.I3) BARD1 - 3; (Job.2.S4.I4) BARD1 - 3 : **3.0**

**Job # 3**
(Job.3.S1.I1) BARD1 - 2; (Job.3.S1.I2) BARD1 - 2; (Job.3.S1.I3) BARD1 - 2; (Job.3.S1.I4) BARD1 - 2 : **2.0** (Job.3.S2.I1) BARD1 - 3; (Job.3.S2.I2) BARD1 - 89; (Job.3.S2.I3) BARD1 - 84; (Job.3.S2.I4) BARD1 - 84 : **84.0** (Job.3.S3.I1) BARD1 - 3; (Job.3.S3.I2) BARD1 - 3; (Job.3.S3.I3) BARD1 - 3; (Job.3.S3.I4) BARD1 - 3 : **3.0** (Job.3.S4.I1)

BARD1 - 8; (Job.3.S4.I2) BARD1 - 4; (Job.3.S4.I3) BARD1 - 6; (Job.3.S4.I4) BARD1 - 6 : **6.0**

**Job # 4**
(Job.4.S1.I1) BARD1 - 4; (Job.4.S1.I2) BARD1 - 3; (Job.4.S1.I3) BARD1 - 4; (Job.4.S1.I4) BARD1 - 3 : **3.5** (Job.4.S2.I1) BARD1 - 15; (Job.4.S2.I2) BARD1 - 64; (Job.4.S2.I3) BARD1 - 10; (Job.4.S2.I4) BARD1 - 73 : **39.5** (Job.4.S3.I1) BARD1 - 10; (Job.4.S3.I2) BARD1 - 6; (Job.4.S3.I3) BARD1 - 9; (Job.4.S3.I4) BARD1 - 10 : **9.5** (Job.4.S4.I1) BARD1 - 7; (Job.4.S4.I2) BARD1 - 7; (Job.4.S4.I3) BARD1 - 3; (Job.4.S4.I4) BARD1 - 5 : **6.0**

*Best ranks of each set–island*: 3 - 3 - 2 - 10 - 5 - 3 - 3 - 5 - 2 - 5 - 4 - 3 - 2 - 3 - 3 - 4 - 3 - 10 - 6 - 3 -
*Median*: 5.0
*Median of best*: 3.0

### A.1.4 BRCA1

**Job # 0**
(Job.0.S1.I1) BRCA1 - 6; (Job.0.S1.I2) BRCA1 - 6; (Job.0.S1.I3) BRCA1 - 6; (Job.0.S1.I4) BRCA1 - 6 : **6.0** (Job.0.S2.I1) BRCA1 - 9; (Job.0.S2.I2) BRCA1 - 9; (Job.0.S2.I3) BRCA1 - 9; (Job.0.S2.I4) BRCA1 - 10 : **9.0** (Job.0.S3.I1) BRCA1 - 7; (Job.0.S3.I2) BRCA1 - 5; (Job.0.S3.I3) BRCA1 - 13; (Job.0.S3.I4) BRCA1 - 6 : **6.5** (Job.0.S4.I1) BRCA1 - 14; (Job.0.S4.I2) BRCA1 - 4; (Job.0.S4.I3) BRCA1 - 4; (Job.0.S4.I4) BRCA1 - 4 : **4.0**

**Job # 1**
(Job.1.S1.I1) BRCA1 - 12; (Job.1.S1.I2) BRCA1 - 16; (Job.1.S1.I3) BRCA1 - 8; (Job.1.S1.I4) BRCA1 - 19 : **14.0** (Job.1.S2.I1) BRCA1 - 13; (Job.1.S2.I2) BRCA1 - 7; (Job.1.S2.I3) BRCA1 - 7; (Job.1.S2.I4) BRCA1 - 7 : **7.0** (Job.1.S3.I1) BRCA1 - 11; (Job.1.S3.I2) BRCA1 - 10; (Job.1.S3.I3) BRCA1 - 11; (Job.1.S3.I4) BRCA1 - 10 : **10.5** (Job.1.S4.I1) BRCA1 - 5; (Job.1.S4.I2) BRCA1 - 5; (Job.1.S4.I3) BRCA1 - 5; (Job.1.S4.I4) BRCA1 - 5 : **5.0**

**Job # 2**
(Job.2.S1.I1) BRCA1 - 6; (Job.2.S1.I2) BRCA1 - 6; (Job.2.S1.I3) BRCA1 - 6; (Job.2.S1.I4)

BRCA1 - 6 : **6.0** (Job.2.S2.I1) BRCA1 - 6; (Job.2.S2.I2) BRCA1 - 4; (Job.2.S2.I3) BRCA1 - 7; (Job.2.S2.I4) BRCA1 - 4 : **5.0** (Job.2.S3.I1) BRCA1 - 3; (Job.2.S3.I2) BRCA1 - 6; (Job.2.S3.I3) BRCA1 - 3; (Job.2.S3.I4) BRCA1 - 3 : **3.0** (Job.2.S4.I1) BRCA1 - 13; (Job.2.S4.I2) BRCA1 - 11; (Job.2.S4.I3) BRCA1 - 12; (Job.2.S4.I4) BRCA1 - 16 : **12.5**

**Job # 3**

(Job.3.S1.I1) BRCA1 - 3; (Job.3.S1.I2) BRCA1 - 3; (Job.3.S1.I3) BRCA1 - 3; (Job.3.S1.I4) BRCA1 - 3 : **3.0** (Job.3.S2.I1) BRCA1 - 12; (Job.3.S2.I2) BRCA1 - 12; (Job.3.S2.I3) BRCA1 - 12; (Job.3.S2.I4) BRCA1 - 12 : **12.0** (Job.3.S3.I1) BRCA1 - 3; (Job.3.S3.I2) BRCA1 - 2; (Job.3.S3.I3) BRCA1 - 14; (Job.3.S3.I4) BRCA1 - 3 : **3.0** (Job.3.S4.I1) BRCA1 - 41; (Job.3.S4.I2) BRCA1 - 102; (Job.3.S4.I3) BRCA1 - 108; (Job.3.S4.I4) BRCA1 - 105 : **103.5**

**Job # 4**

(Job.4.S1.I1) BRCA1 - 18; (Job.4.S1.I2) BRCA1 - 33; (Job.4.S1.I3) BRCA1 - 53; (Job.4.S1.I4) BRCA1 - 22 : **27.5** (Job.4.S2.I1) BRCA1 - 21; (Job.4.S2.I2) BRCA1 - 6; (Job.4.S2.I3) BRCA1 - 13; (Job.4.S2.I4) BRCA1 - 18 : **15.5** (Job.4.S3.I1) BRCA1 - 6; (Job.4.S3.I2) BRCA1 - 6; (Job.4.S3.I3) BRCA1 - 6; (Job.4.S3.I4) BRCA1 - 6 : **6.0** (Job.4.S4.I1) BRCA1 - 7; (Job.4.S4.I2) BRCA1 - 9; (Job.4.S4.I3) BRCA1 - 9; (Job.4.S4.I4) BRCA1 - 7 : **8.0**

*Best ranks of each set–island*: 6 - 9 - 5 - 4 - 8 - 7 - 10 - 5 - 6 - 4 - 3 - 11 - 3 - 12 - 2 - 41 - 18 - 6 - 6 - 7 -
*Median*: 6.75
*Median of best*: 6.0

### A.1.5 BRCA2

**Job # 0**

(Job.0.S1.I1) BRCA2 - 129; (Job.0.S1.I2) BRCA2 - 129; (Job.0.S1.I3) BRCA2 - 129; (Job.0.S1.I4) BRCA2 - 128 : **129.0** (Job.0.S2.I1) BRCA2 - 77; (Job.0.S2.I2) BRCA2 - 98; (Job.0.S2.I3) BRCA2 - 102; (Job.0.S2.I4) BRCA2 - 91 : **94.5** (Job.0.S3.I1) BRCA2 - 128; (Job.0.S3.I2) BRCA2 - 127; (Job.0.S3.I3) BRCA2 - 128; (Job.0.S3.I4) BRCA2 - 127 : **127.5** (Job.0.S4.I1) BRCA2 - 185; (Job.0.S4.I2) BRCA2 - 190; (Job.0.S4.I3) BRCA2 - 130; (Job.0.S4.I4) BRCA2 - 187 : **186.0**

**Job # 1**

(Job.1.S1.I1) BRCA2 - 130; (Job.1.S1.I2) BRCA2 - 121; (Job.1.S1.I3) BRCA2 - 121; (Job.1.S1.I4) BRCA2 - 121 : **121.0** (Job.1.S2.I1) BRCA2 - 192; (Job.1.S2.I2) BRCA2 - 118; (Job.1.S2.I3) BRCA2 - 212; (Job.1.S2.I4) BRCA2 - 265 : **202.0** (Job.1.S3.I1) BRCA2 - 482; (Job.1.S3.I2) BRCA2 - 473; (Job.1.S3.I3) BRCA2 - 161; (Job.1.S3.I4) BRCA2 - 477 : **475.0** (Job.1.S4.I1) BRCA2 - 135; (Job.1.S4.I2) BRCA2 - 134; (Job.1.S4.I3) BRCA2 - 135; (Job.1.S4.I4) BRCA2 - 135 : **135.0**

**Job # 2**

(Job.2.S1.I1) BRCA2 - 169; (Job.2.S1.I2) BRCA2 - 185; (Job.2.S1.I3) BRCA2 - 167; (Job.2.S1.I4) BRCA2 - 161 : **168.0** (Job.2.S2.I1) BRCA2 - 112; (Job.2.S2.I2) BRCA2 - 117; (Job.2.S2.I3) BRCA2 - 127; (Job.2.S2.I4) BRCA2 - 127 : **122.0** (Job.2.S3.I1) BRCA2 - 119; (Job.2.S3.I2) BRCA2 - 204; (Job.2.S3.I3) BRCA2 - 241; (Job.2.S3.I4) BRCA2 - 195 : **199.5** (Job.2.S4.I1) BRCA2 - 119; (Job.2.S4.I2) BRCA2 - 143; (Job.2.S4.I3) BRCA2 - 143; (Job.2.S4.I4) BRCA2 - 130 : **136.5**

**Job # 3**

(Job.3.S1.I1) BRCA2 - 151; (Job.3.S1.I2) BRCA2 - 151; (Job.3.S1.I3) BRCA2 - 22; (Job.3.S1.I4) BRCA2 - 161 : **151.0** (Job.3.S2.I1) BRCA2 - 173; (Job.3.S2.I2) BRCA2 - 194; (Job.3.S2.I3) BRCA2 - 182; (Job.3.S2.I4) BRCA2 - 191 : **186.5** (Job.3.S3.I1) BRCA2 - 189; (Job.3.S3.I2) BRCA2 - 557; (Job.3.S3.I3) BRCA2 - 194; (Job.3.S3.I4) BRCA2 - 194 : **194.0** (Job.3.S4.I1) BRCA2 - 207; (Job.3.S4.I2) BRCA2 - 222; (Job.3.S4.I3) BRCA2 - 341; (Job.3.S4.I4) BRCA2 - 183 : **214.5**

**Job # 4**

(Job.4.S1.I1) BRCA2 - 174; (Job.4.S1.I2) BRCA2 - 178; (Job.4.S1.I3) BRCA2 - 188; (Job.4.S1.I4) BRCA2 - 184 : **181.0** (Job.4.S2.I1) BRCA2 - 136; (Job.4.S2.I2) BRCA2 - 130; (Job.4.S2.I3) BRCA2 - 137; (Job.4.S2.I4) BRCA2 - 136 : **136.0** (Job.4.S3.I1) BRCA2 - 121; (Job.4.S3.I2) BRCA2 - 122; (Job.4.S3.I3) BRCA2 - 140; (Job.4.S3.I4) BRCA2 - 122 : **122.0** (Job.4.S4.I1) BRCA2 - 76; (Job.4.S4.I2) BRCA2 - 72; (Job.4.S4.I3) BRCA2 - 98; (Job.4.S4.I4) BRCA2 - 72 : **74.0**

*Best ranks of each set–island*: 128 - 77 - 127 - 130 - 121 - 118 - 161 - 134 - 161 - 112 - 119 - 119 - 22 - 173 - 189 - 183 - 174 - 130 - 121 - 72 -
*Median*: 143.75

*Median of best*: 127.5

### A.1.6 CASP8

**Job # 0**

(Job.0.S1.I1) CASP8 - 24; (Job.0.S1.I2) CASP8 - 21; (Job.0.S1.I3) CASP8 - 27; (Job.0.S1.I4) CASP8 - 28 : **25.5** (Job.0.S2.I1) CASP8 - 31; (Job.0.S2.I2) CASP8 - 30; (Job.0.S2.I3) CASP8 - 30; (Job.0.S2.I4) CASP8 - 38 : **30.5** (Job.0.S3.I1) CASP8 - 47; (Job.0.S3.I2) CASP8 - 50; (Job.0.S3.I3) CASP8 - 47; (Job.0.S3.I4) CASP8 - 55 : **48.5** (Job.0.S4.I1) CASP8 - 88; (Job.0.S4.I2) CASP8 - 98; (Job.0.S4.I3) CASP8 - 98; (Job.0.S4.I4) CASP8 - 98 : **98.0**

**Job # 1**

(Job.1.S1.I1) CASP8 - 29; (Job.1.S1.I2) CASP8 - 23; (Job.1.S1.I3) CASP8 - 29; (Job.1.S1.I4) CASP8 - 29 : **29.0** (Job.1.S2.I1) CASP8 - 103; (Job.1.S2.I2) CASP8 - 30; (Job.1.S2.I3) CASP8 - 29; (Job.1.S2.I4) CASP8 - 108 : **66.5** (Job.1.S3.I1) CASP8 - 52; (Job.1.S3.I2) CASP8 - 32; (Job.1.S3.I3) CASP8 - 49; (Job.1.S3.I4) CASP8 - 35 : **42.0** (Job.1.S4.I1) CASP8 - 44; (Job.1.S4.I2) CASP8 - 45; (Job.1.S4.I3) CASP8 - 45; (Job.1.S4.I4) CASP8 - 45 : **45.0**

**Job # 2**

(Job.2.S1.I1) CASP8 - 38; (Job.2.S1.I2) CASP8 - 60; (Job.2.S1.I3) CASP8 - 38; (Job.2.S1.I4) CASP8 - 38 : **38.0** (Job.2.S2.I1) CASP8 - 42; (Job.2.S2.I2) CASP8 - 38; (Job.2.S2.I3) CASP8 - 18; (Job.2.S2.I4) CASP8 - 16 : **28.0** (Job.2.S3.I1) CASP8 - 43; (Job.2.S3.I2) CASP8 - 41; (Job.2.S3.I3) CASP8 - 32; (Job.2.S3.I4) CASP8 - 43 : **42.0** (Job.2.S4.I1) CASP8 - 4; (Job.2.S4.I2) CASP8 - 5; (Job.2.S4.I3) CASP8 - 7; (Job.2.S4.I4) CASP8 - 5 : **5.0**

**Job # 3**

(Job.3.S1.I1) CASP8 - 50; (Job.3.S1.I2) CASP8 - 49; (Job.3.S1.I3) CASP8 - 50; (Job.3.S1.I4) CASP8 - 50 : **50.0** (Job.3.S2.I1) CASP8 - 26; (Job.3.S2.I2) CASP8 - 25; (Job.3.S2.I3) CASP8 - 26; (Job.3.S2.I4) CASP8 - 23 : **25.5** (Job.3.S3.I1) CASP8 - 45; (Job.3.S3.I2) CASP8 - 55; (Job.3.S3.I3) CASP8 - 44; (Job.3.S3.I4) CASP8 - 34 : **44.5** (Job.3.S4.I1) CASP8 - 8; (Job.3.S4.I2) CASP8 - 13; (Job.3.S4.I3) CASP8 - 10; (Job.3.S4.I4) CASP8 - 15 : **11.5**

**Job # 4**
(Job.4.S1.I1) CASP8 - 53; (Job.4.S1.I2) CASP8 - 54; (Job.4.S1.I3) CASP8 - 50; (Job.4.S1.I4) CASP8 - 55 : **53.5** (Job.4.S2.I1) CASP8 - 56; (Job.4.S2.I2) CASP8 - 42; (Job.4.S2.I3) CASP8 - 46; (Job.4.S2.I4) CASP8 - 44 : **45.0** (Job.4.S3.I1) CASP8 - 40; (Job.4.S3.I2) CASP8 - 32; (Job.4.S3.I3) CASP8 - 32; (Job.4.S3.I4) CASP8 - 39 : **35.5** (Job.4.S4.I1) CASP8 - 84; (Job.4.S4.I2) CASP8 - 64; (Job.4.S4.I3) CASP8 - 91; (Job.4.S4.I4) CASP8 - 115 : **87.5**

*Best ranks of each set–island*: 21 - 30 - 47 - 88 - 23 - 29 - 32 - 44 - 38 - 16 - 32 - 4 - 49 - 23 - 34 - 8 - 50 - 42 - 32 - 64 -
*Median*: 42.0
*Median of best*: 32.0

### A.1.7 CHEK2

**Job # 0**
(Job.0.S1.I1) CHEK2 - 47; (Job.0.S1.I2) CHEK2 - 45; (Job.0.S1.I3) CHEK2 - 45; (Job.0.S1.I4) CHEK2 - 50 : **46.0** (Job.0.S2.I1) CHEK2 - 74; (Job.0.S2.I2) CHEK2 - 76; (Job.0.S2.I3) CHEK2 - 76; (Job.0.S2.I4) CHEK2 - 74 : **75.0** (Job.0.S3.I1) CHEK2 - 67; (Job.0.S3.I2) CHEK2 - 69; (Job.0.S3.I3) CHEK2 - 65; (Job.0.S3.I4) CHEK2 - 62 : **66.0** (Job.0.S4.I1) CHEK2 - 153; (Job.0.S4.I2) CHEK2 - 138; (Job.0.S4.I3) CHEK2 - 153; (Job.0.S4.I4) CHEK2 - 151 : **152.0**

**Job # 1**
(Job.1.S1.I1) CHEK2 - 79; (Job.1.S1.I2) CHEK2 - 54; (Job.1.S1.I3) CHEK2 - 65; (Job.1.S1.I4) CHEK2 - 65 : **65.0** (Job.1.S2.I1) CHEK2 - 85; (Job.1.S2.I2) CHEK2 - 88; (Job.1.S2.I3) CHEK2 - 83; (Job.1.S2.I4) CHEK2 - 88 : **86.5** (Job.1.S3.I1) CHEK2 - 107; (Job.1.S3.I2) CHEK2 - 121; (Job.1.S3.I3) CHEK2 - 124; (Job.1.S3.I4) CHEK2 - 107 : **114.0** (Job.1.S4.I1) CHEK2 - 63; (Job.1.S4.I2) CHEK2 - 70; (Job.1.S4.I3) CHEK2 - 63; (Job.1.S4.I4) CHEK2 - 63 : **63.0**

**Job # 2**
(Job.2.S1.I1) CHEK2 - 91; (Job.2.S1.I2) CHEK2 - 111; (Job.2.S1.I3) CHEK2 - 111; (Job.2.S1.I4) CHEK2 - 116 : **111.0** (Job.2.S2.I1) CHEK2 - 84; (Job.2.S2.I2) CHEK2 - 83; (Job.2.S2.I3) CHEK2 - 84; (Job.2.S2.I4) CHEK2 - 83 : **83.5** (Job.2.S3.I1) CHEK2 - 68; (Job.2.S3.I2) CHEK2 - 67; (Job.2.S3.I3) CHEK2 - 68; (Job.2.S3.I4) CHEK2 - 67

: **67.5** (Job.2.S4.I1) CHEK2 - 99; (Job.2.S4.I2) CHEK2 - 145; (Job.2.S4.I3) CHEK2 - 128; (Job.2.S4.I4) CHEK2 - 82 : **113.5**

**Job # 3**

(Job.3.S1.I1) CHEK2 - 68; (Job.3.S1.I2) CHEK2 - 70; (Job.3.S1.I3) CHEK2 - 72; (Job.3.S1.I4) CHEK2 - 68 : **69.0** (Job.3.S2.I1) CHEK2 - 63; (Job.3.S2.I2) CHEK2 - 63; (Job.3.S2.I3) CHEK2 - 74; (Job.3.S2.I4) CHEK2 - 63 : **63.0** (Job.3.S3.I1) CHEK2 - 78; (Job.3.S3.I2) CHEK2 - 100; (Job.3.S3.I3) CHEK2 - 55; (Job.3.S3.I4) CHEK2 - 90 : **84.0** (Job.3.S4.I1) CHEK2 - 47; (Job.3.S4.I2) CHEK2 - 76; (Job.3.S4.I3) CHEK2 - 2; (Job.3.S4.I4) CHEK2 - 1 : **24.5**

**Job # 4**

(Job.4.S1.I1) CHEK2 - 69; (Job.4.S1.I2) CHEK2 - 70; (Job.4.S1.I3) CHEK2 - 71; (Job.4.S1.I4) CHEK2 - 68 : **69.5** (Job.4.S2.I1) CHEK2 - 74; (Job.4.S2.I2) CHEK2 - 71; (Job.4.S2.I3) CHEK2 - 73; (Job.4.S2.I4) CHEK2 - 68 : **72.0** (Job.4.S3.I1) CHEK2 - 72; (Job.4.S3.I2) CHEK2 - 121; (Job.4.S3.I3) CHEK2 - 59; (Job.4.S3.I4) CHEK2 - 68 : **70.0** (Job.4.S4.I1) CHEK2 - 77; (Job.4.S4.I2) CHEK2 - 76; (Job.4.S4.I3) CHEK2 - 76; (Job.4.S4.I4) CHEK2 - 77 : **76.5**

*Best ranks of each set–island*: 45 - 74 - 62 - 138 - 54 - 83 - 107 - 63 - 91 - 83 - 67 - 82 - 68 - 63 - 55 - 1 - 68 - 68 - 59 - 76 -
*Median*: 71.0
*Median of best*: 68.0

### A.1.8 NCOA3

**Job # 0**

(Job.0.S1.I1) NCOA3 - 15; (Job.0.S1.I2) NCOA3 - 13; (Job.0.S1.I3) NCOA3 - 25; (Job.0.S1.I4) NCOA3 - 25 : **20.0** (Job.0.S2.I1) NCOA3 - 4; (Job.0.S2.I2) NCOA3 - 17; (Job.0.S2.I3) NCOA3 - 18; (Job.0.S2.I4) NCOA3 - 4 : **10.5** (Job.0.S3.I1) NCOA3 - 38; (Job.0.S3.I2) NCOA3 - 40; (Job.0.S3.I3) NCOA3 - 40; (Job.0.S3.I4) NCOA3 - 24 : **39.0** (Job.0.S4.I1) NCOA3 - 14; (Job.0.S4.I2) NCOA3 - 14; (Job.0.S4.I3) NCOA3 - 14; (Job.0.S4.I4) NCOA3 - 14 : **14.0**

**Job # 1**

(Job.1.S1.I1) NCOA3 - 36; (Job.1.S1.I2) NCOA3 - 35; (Job.1.S1.I3) NCOA3 - 36;

(Job.1.S1.I4) NCOA3 - 34 : **35.5** (Job.1.S2.I1) NCOA3 - 18; (Job.1.S2.I2) NCOA3 - 20; (Job.1.S2.I3) NCOA3 - 18; (Job.1.S2.I4) NCOA3 - 21 : **19.0** (Job.1.S3.I1) NCOA3 - 36; (Job.1.S3.I2) NCOA3 - 77; (Job.1.S3.I3) NCOA3 - 27; (Job.1.S3.I4) NCOA3 - 33 : **34.5** (Job.1.S4.I1) NCOA3 - 18; (Job.1.S4.I2) NCOA3 - 16; (Job.1.S4.I3) NCOA3 - 16; (Job.1.S4.I4) NCOA3 - 17 : **16.5**

**Job # 2**

(Job.2.S1.I1) NCOA3 - 45; (Job.2.S1.I2) NCOA3 - 31; (Job.2.S1.I3) NCOA3 - 31; (Job.2.S1.I4) NCOA3 - 48 : **38.0** (Job.2.S2.I1) NCOA3 - 27; (Job.2.S2.I2) NCOA3 - 34; (Job.2.S2.I3) NCOA3 - 27; (Job.2.S2.I4) NCOA3 - 27 : **27.0** (Job.2.S3.I1) NCOA3 - 3; (Job.2.S3.I2) NCOA3 - 3; (Job.2.S3.I3) NCOA3 - 3; (Job.2.S3.I4) NCOA3 - 3 : **3.0** (Job.2.S4.I1) NCOA3 - 34; (Job.2.S4.I2) NCOA3 - 29; (Job.2.S4.I3) NCOA3 - 34; (Job.2.S4.I4) NCOA3 - 39 : **34.0**

**Job # 3**

(Job.3.S1.I1) NCOA3 - 15; (Job.3.S1.I2) NCOA3 - 15; (Job.3.S1.I3) NCOA3 - 15; (Job.3.S1.I4) NCOA3 - 15 : **15.0** (Job.3.S2.I1) NCOA3 - 28; (Job.3.S2.I2) NCOA3 - 39; (Job.3.S2.I3) NCOA3 - 20; (Job.3.S2.I4) NCOA3 - 39 : **33.5** (Job.3.S3.I1) NCOA3 - 34; (Job.3.S3.I2) NCOA3 - 27; (Job.3.S3.I3) NCOA3 - 33; (Job.3.S3.I4) NCOA3 - 31 : **32.0** (Job.3.S4.I1) NCOA3 - 44; (Job.3.S4.I2) NCOA3 - 48; (Job.3.S4.I3) NCOA3 - 53; (Job.3.S4.I4) NCOA3 - 34 : **46.0**

**Job # 4**

(Job.4.S1.I1) NCOA3 - 4; (Job.4.S1.I2) NCOA3 - 4; (Job.4.S1.I3) NCOA3 - 2; (Job.4.S1.I4) NCOA3 - 4 : **4.0** (Job.4.S2.I1) NCOA3 - 21; (Job.4.S2.I2) NCOA3 - 21; (Job.4.S2.I3) NCOA3 - 18; (Job.4.S2.I4) NCOA3 - 25 : **21.0** (Job.4.S3.I1) NCOA3 - 42; (Job.4.S3.I2) NCOA3 - 48; (Job.4.S3.I3) NCOA3 - 46; (Job.4.S3.I4) NCOA3 - 45 : **45.5** (Job.4.S4.I1) NCOA3 - 8; (Job.4.S4.I2) NCOA3 - 8; (Job.4.S4.I3) NCOA3 - 6; (Job.4.S4.I4) NCOA3 - 13 : **8.0**

*Best ranks of each set–island*: 13 - 4 - 24 - 14 - 34 - 18 - 27 - 16 - 31 - 27 - 3 - 29 - 15 - 20 - 27 - 34 - 2 - 18 - 42 - 6 -
*Median*: 24.0
*Median of best*: 19.0

### A.1.9 PIK3CA

**Job # 0**

(Job.0.S1.I1) PIK3CA - 28; (Job.0.S1.I2) PIK3CA - 28; (Job.0.S1.I3) PIK3CA - 28; (Job.0.S1.I4) PIK3CA - 28 : **28.0** (Job.0.S2.I1) PIK3CA - 25; (Job.0.S2.I2) PIK3CA - 19; (Job.0.S2.I3) PIK3CA - 22; (Job.0.S2.I4) PIK3CA - 18 : **20.5** (Job.0.S3.I1) PIK3CA - 3; (Job.0.S3.I2) PIK3CA - 3; (Job.0.S3.I3) PIK3CA - 3; (Job.0.S3.I4) PIK3CA - 3 : **3.0** (Job.0.S4.I1) PIK3CA - 16; (Job.0.S4.I2) PIK3CA - 14; (Job.0.S4.I3) PIK3CA - 16; (Job.0.S4.I4) PIK3CA - 16 : **16.0**

**Job # 1**

(Job.1.S1.I1) PIK3CA - 6; (Job.1.S1.I2) PIK3CA - 6; (Job.1.S1.I3) PIK3CA - 6; (Job.1.S1.I4) PIK3CA - 6 : **6.0** (Job.1.S2.I1) PIK3CA - 6; (Job.1.S2.I2) PIK3CA - 5; (Job.1.S2.I3) PIK3CA - 6; (Job.1.S2.I4) PIK3CA - 6 : **6.0** (Job.1.S3.I1) PIK3CA - 25; (Job.1.S3.I2) PIK3CA - 29; (Job.1.S3.I3) PIK3CA - 25; (Job.1.S3.I4) PIK3CA - 25 : **25.0** (Job.1.S4.I1) PIK3CA - 12; (Job.1.S4.I2) PIK3CA - 13; (Job.1.S4.I3) PIK3CA - 12; (Job.1.S4.I4) PIK3CA - 12 : **12.0**

**Job # 2**

(Job.2.S1.I1) PIK3CA - 18; (Job.2.S1.I2) PIK3CA - 9; (Job.2.S1.I3) PIK3CA - 18; (Job.2.S1.I4) PIK3CA - 14 : **16.0** (Job.2.S2.I1) PIK3CA - 3; (Job.2.S2.I2) PIK3CA - 3; (Job.2.S2.I3) PIK3CA - 2; (Job.2.S2.I4) PIK3CA - 2 : **2.5** (Job.2.S3.I1) PIK3CA - 6; (Job.2.S3.I2) PIK3CA - 4; (Job.2.S3.I3) PIK3CA - 6; (Job.2.S3.I4) PIK3CA - 4 : **5.0** (Job.2.S4.I1) PIK3CA - 14; (Job.2.S4.I2) PIK3CA - 16; (Job.2.S4.I3) PIK3CA - 16; (Job.2.S4.I4) PIK3CA - 8 : **15.0**

**Job # 3**

(Job.3.S1.I1) PIK3CA - 5; (Job.3.S1.I2) PIK3CA - 9; (Job.3.S1.I3) PIK3CA - 6; (Job.3.S1.I4) PIK3CA - 9 : **7.5** (Job.3.S2.I1) PIK3CA - 5; (Job.3.S2.I2) PIK3CA - 4; (Job.3.S2.I3) PIK3CA - 4; (Job.3.S2.I4) PIK3CA - 5 : **4.5** (Job.3.S3.I1) PIK3CA - 29; (Job.3.S3.I2) PIK3CA - 24; (Job.3.S3.I3) PIK3CA - 29; (Job.3.S3.I4) PIK3CA - 29 : **29.0** (Job.3.S4.I1) PIK3CA - 19; (Job.3.S4.I2) PIK3CA - 14; (Job.3.S4.I3) PIK3CA - 16; (Job.3.S4.I4) PIK3CA - 19 : **17.5**

**Job # 4**

(Job.4.S1.I1) PIK3CA - 1; (Job.4.S1.I2) PIK3CA - 3; (Job.4.S1.I3) PIK3CA - 1; (Job.4.S1.I4) PIK3CA - 1 : **1.0** (Job.4.S2.I1) PIK3CA - 17; (Job.4.S2.I2) PIK3CA

- 13; (Job.4.S2.I3) PIK3CA - 13; (Job.4.S2.I4) PIK3CA - 14 : **13.5** (Job.4.S3.I1) PIK3CA - 1; (Job.4.S3.I2) PIK3CA - 3; (Job.4.S3.I3) PIK3CA - 6; (Job.4.S3.I4) PIK3CA - 3 : **3.0** (Job.4.S4.I1) PIK3CA - 6; (Job.4.S4.I2) PIK3CA - 6; (Job.4.S4.I3) PIK3CA - 6; (Job.4.S4.I4) PIK3CA - 6 : **6.0**

*Best ranks of each set–island*: 28 - 18 - 3 - 14 - 6 - 5 - 25 - 12 - 9 - 2 - 4 - 8 - 5 - 4 - 24 - 14 - 1 - 13 - 1 - 6 -
*Median*: 9.75
*Median of best*: 7.0

### A.1.10 PPM1D

**Job # 0**
(Job.0.S1.I1) PPM1D - 215; (Job.0.S1.I2) PPM1D - 169; (Job.0.S1.I3) PPM1D - 224; (Job.0.S1.I4) PPM1D - 231 : **219.5** (Job.0.S2.I1) PPM1D - 249; (Job.0.S2.I2) PPM1D - 308; (Job.0.S2.I3) PPM1D - 234; (Job.0.S2.I4) PPM1D - 236 : **242.5** (Job.0.S3.I1) PPM1D - 162; (Job.0.S3.I2) PPM1D - 160; (Job.0.S3.I3) PPM1D - 161; (Job.0.S3.I4) PPM1D - 160 : **160.5** (Job.0.S4.I1) PPM1D - 530; (Job.0.S4.I2) PPM1D - 530; (Job.0.S4.I3) PPM1D - 546; (Job.0.S4.I4) PPM1D - 546 : **538.0**

**Job # 1**
(Job.1.S1.I1) PPM1D - 415; (Job.1.S1.I2) PPM1D - 162; (Job.1.S1.I3) PPM1D - 161; (Job.1.S1.I4) PPM1D - 416 : **288.5** (Job.1.S2.I1) PPM1D - 932; (Job.1.S2.I2) PPM1D - 355; (Job.1.S2.I3) PPM1D - 932; (Job.1.S2.I4) PPM1D - 879 : **905.5** (Job.1.S3.I1) PPM1D - 595; (Job.1.S3.I2) PPM1D - 603; (Job.1.S3.I3) PPM1D - 595; (Job.1.S3.I4) PPM1D - 572 : **595.0** (Job.1.S4.I1) PPM1D - 259; (Job.1.S4.I2) PPM1D - 589; (Job.1.S4.I3) PPM1D - 513; (Job.1.S4.I4) PPM1D - 329 : **421.0**

**Job # 2**
(Job.2.S1.I1) PPM1D - 218; (Job.2.S1.I2) PPM1D - 196; (Job.2.S1.I3) PPM1D - 207; (Job.2.S1.I4) PPM1D - 207 : **207.0** (Job.2.S2.I1) PPM1D - 226; (Job.2.S2.I2) PPM1D - 441; (Job.2.S2.I3) PPM1D - 400; (Job.2.S2.I4) PPM1D - 227 : **313.5** (Job.2.S3.I1) PPM1D - 1065; (Job.2.S3.I2) PPM1D - 1686; (Job.2.S3.I3) PPM1D - 1618; (Job.2.S3.I4) PPM1D - 1407 : **1512.5** (Job.2.S4.I1) PPM1D - 159; (Job.2.S4.I2) PPM1D - 162; (Job.2.S4.I3) PPM1D - 165; (Job.2.S4.I4) PPM1D - 165 : **163.5**

**Job # 3**
(Job.3.S1.I1) PPM1D - 330; (Job.3.S1.I2) PPM1D - 321; (Job.3.S1.I3) PPM1D - 330; (Job.3.S1.I4) PPM1D - 213 : **325.5** (Job.3.S2.I1) PPM1D - 296; (Job.3.S2.I2) PPM1D - 191; (Job.3.S2.I3) PPM1D - 270; (Job.3.S2.I4) PPM1D - 1054 : **283.0** (Job.3.S3.I1) PPM1D - 2; (Job.3.S3.I2) PPM1D - 2; (Job.3.S3.I3) PPM1D - 2; (Job.3.S3.I4) PPM1D - 2 : **2.0** (Job.3.S4.I1) PPM1D - 167; (Job.3.S4.I2) PPM1D - 215; (Job.3.S4.I3) PPM1D - 175; (Job.3.S4.I4) PPM1D - 161 : **171.0**

**Job # 4**
(Job.4.S1.I1) PPM1D - 1718; (Job.4.S1.I2) PPM1D - 1699; (Job.4.S1.I3) PPM1D - 1722; (Job.4.S1.I4) PPM1D - 1721 : **1719.5** (Job.4.S2.I1) PPM1D - 125; (Job.4.S2.I2) PPM1D - 125; (Job.4.S2.I3) PPM1D - 125; (Job.4.S2.I4) PPM1D - 172 : **125.0** (Job.4.S3.I1) PPM1D - 182; (Job.4.S3.I2) PPM1D - 199; (Job.4.S3.I3) PPM1D - 230; (Job.4.S3.I4) PPM1D - 214 : **206.5** (Job.4.S4.I1) PPM1D - 171; (Job.4.S4.I2) PPM1D - 171; (Job.4.S4.I3) PPM1D - 191; (Job.4.S4.I4) PPM1D - 189 : **180.0**

*Best ranks of each set–island*: 169 - 234 - 160 - 530 - 161 - 355 - 572 - 259 - 196 - 226 - 1065 - 159 - 213 - 191 - 2 - 161 - 1699 - 125 - 182 - 171 -
*Median*: 262.75
*Median of best*: 193.5

### A.1.11 PTEN

**Job # 0**
(Job.0.S1.I1) PTEN - 165; (Job.0.S1.I2) PTEN - 165; (Job.0.S1.I3) PTEN - 134; (Job.0.S1.I4) PTEN - 174 : **165.0** (Job.0.S2.I1) PTEN - 130; (Job.0.S2.I2) PTEN - 130; (Job.0.S2.I3) PTEN - 211; (Job.0.S2.I4) PTEN - 172 : **151.0** (Job.0.S3.I1) PTEN - 138; (Job.0.S3.I2) PTEN - 139; (Job.0.S3.I3) PTEN - 139; (Job.0.S3.I4) PTEN - 141 : **139.0** (Job.0.S4.I1) PTEN - 139; (Job.0.S4.I2) PTEN - 146; (Job.0.S4.I3) PTEN - 130; (Job.0.S4.I4) PTEN - 128 : **134.5**

**Job # 1**
(Job.1.S1.I1) PTEN - 93; (Job.1.S1.I2) PTEN - 94; (Job.1.S1.I3) PTEN - 94; (Job.1.S1.I4) PTEN - 92 : **93.5** (Job.1.S2.I1) PTEN - 107; (Job.1.S2.I2) PTEN - 107; (Job.1.S2.I3) PTEN - 107; (Job.1.S2.I4) PTEN - 111 : **107.0** (Job.1.S3.I1) PTEN - 141; (Job.1.S3.I2) PTEN - 140; (Job.1.S3.I3) PTEN - 141; (Job.1.S3.I4) PTEN - 140 : **140.5** (Job.1.S4.I1)

PTEN - 214; (Job.1.S4.I2) PTEN - 149; (Job.1.S4.I3) PTEN - 138; (Job.1.S4.I4) PTEN - 142 : **145.5**

**Job # 2**
(Job.2.S1.I1) PTEN - 141; (Job.2.S1.I2) PTEN - 138; (Job.2.S1.I3) PTEN - 142; (Job.2.S1.I4) PTEN - 143 : **141.5** (Job.2.S2.I1) PTEN - 193; (Job.2.S2.I2) PTEN - 193; (Job.2.S2.I3) PTEN - 143; (Job.2.S2.I4) PTEN - 142 : **168.0** (Job.2.S3.I1) PTEN - 336; (Job.2.S3.I2) PTEN - 428; (Job.2.S3.I3) PTEN - 428; (Job.2.S3.I4) PTEN - 356 : **392.0** (Job.2.S4.I1) PTEN - 151; (Job.2.S4.I2) PTEN - 155; (Job.2.S4.I3) PTEN - 145; (Job.2.S4.I4) PTEN - 151 : **151.0**

**Job # 3**
(Job.3.S1.I1) PTEN - 42; (Job.3.S1.I2) PTEN - 42; (Job.3.S1.I3) PTEN - 43; (Job.3.S1.I4) PTEN - 57 : **42.5** (Job.3.S2.I1) PTEN - 285; (Job.3.S2.I2) PTEN - 280; (Job.3.S2.I3) PTEN - 296; (Job.3.S2.I4) PTEN - 291 : **288.0** (Job.3.S3.I1) PTEN - 108; (Job.3.S3.I2) PTEN - 97; (Job.3.S3.I3) PTEN - 99; (Job.3.S3.I4) PTEN - 100 : **99.5** (Job.3.S4.I1) PTEN - 107; (Job.3.S4.I2) PTEN - 112; (Job.3.S4.I3) PTEN - 119; (Job.3.S4.I4) PTEN - 115 : **113.5**

**Job # 4**
(Job.4.S1.I1) PTEN - 159; (Job.4.S1.I2) PTEN - 159; (Job.4.S1.I3) PTEN - 155; (Job.4.S1.I4) PTEN - 169 : **159.0** (Job.4.S2.I1) PTEN - 767; (Job.4.S2.I2) PTEN - 768; (Job.4.S2.I3) PTEN - 791; (Job.4.S2.I4) PTEN - 791 : **779.5** (Job.4.S3.I1) PTEN - 146; (Job.4.S3.I2) PTEN - 140; (Job.4.S3.I3) PTEN - 139; (Job.4.S3.I4) PTEN - 129 : **139.5** (Job.4.S4.I1) PTEN - 113; (Job.4.S4.I2) PTEN - 216; (Job.4.S4.I3) PTEN - 212; (Job.4.S4.I4) PTEN - 246 : **214.0**

*Best ranks of each set–island*: 134 - 130 - 138 - 128 - 92 - 107 - 140 - 138 - 138 - 142 - 336 - 145 - 42 - 280 - 97 - 107 - 155 - 767 - 129 - 113 -
*Median*: 143.5
*Median of best*: 136.0

### A.1.12 RAD51

**Job # 0**
(Job.0.S1.I1) RAD51 - 12; (Job.0.S1.I2) RAD51 - 35; (Job.0.S1.I3) RAD51 - 35;

(Job.0.S1.I4) RAD51 - 36 : **35.0** (Job.0.S2.I1) RAD51 - 25; (Job.0.S2.I2) RAD51 - 25; (Job.0.S2.I3) RAD51 - 26; (Job.0.S2.I4) RAD51 - 25 : **25.0** (Job.0.S3.I1) RAD51 - 160; (Job.0.S3.I2) RAD51 - 157; (Job.0.S3.I3) RAD51 - 157; (Job.0.S3.I4) RAD51 - 142 : **157.0** (Job.0.S4.I1) RAD51 - 35; (Job.0.S4.I2) RAD51 - 32; (Job.0.S4.I3) RAD51 - 33; (Job.0.S4.I4) RAD51 - 35 : **34.0**

**Job # 1**

(Job.1.S1.I1) RAD51 - 13; (Job.1.S1.I2) RAD51 - 13; (Job.1.S1.I3) RAD51 - 5; (Job.1.S1.I4) RAD51 - 15 : **13.0** (Job.1.S2.I1) RAD51 - 37; (Job.1.S2.I2) RAD51 - 28; (Job.1.S2.I3) RAD51 - 37; (Job.1.S2.I4) RAD51 - 37 : **37.0** (Job.1.S3.I1) RAD51 - 28; (Job.1.S3.I2) RAD51 - 28; (Job.1.S3.I3) RAD51 - 28; (Job.1.S3.I4) RAD51 - 28 : **28.0** (Job.1.S4.I1) RAD51 - 31; (Job.1.S4.I2) RAD51 - 31; (Job.1.S4.I3) RAD51 - 30; (Job.1.S4.I4) RAD51 - 29 : **30.5**

**Job # 2**

(Job.2.S1.I1) RAD51 - 29; (Job.2.S1.I2) RAD51 - 35; (Job.2.S1.I3) RAD51 - 66; (Job.2.S1.I4) RAD51 - 61 : **48.0** (Job.2.S2.I1) RAD51 - 37; (Job.2.S2.I2) RAD51 - 36; (Job.2.S2.I3) RAD51 - 36; (Job.2.S2.I4) RAD51 - 37 : **36.5** (Job.2.S3.I1) RAD51 - 35; (Job.2.S3.I2) RAD51 - 34; (Job.2.S3.I3) RAD51 - 34; (Job.2.S3.I4) RAD51 - 35 : **34.5** (Job.2.S4.I1) RAD51 - 7; (Job.2.S4.I2) RAD51 - 6; (Job.2.S4.I3) RAD51 - 7; (Job.2.S4.I4) RAD51 - 7 : **7.0**

**Job # 3**

(Job.3.S1.I1) RAD51 - 30; (Job.3.S1.I2) RAD51 - 47; (Job.3.S1.I3) RAD51 - 29; (Job.3.S1.I4) RAD51 - 30 : **30.0** (Job.3.S2.I1) RAD51 - 126; (Job.3.S2.I2) RAD51 - 25; (Job.3.S2.I3) RAD51 - 32; (Job.3.S2.I4) RAD51 - 25 : **28.5** (Job.3.S3.I1) RAD51 - 28; (Job.3.S3.I2) RAD51 - 25; (Job.3.S3.I3) RAD51 - 26; (Job.3.S3.I4) RAD51 - 27 : **26.5** (Job.3.S4.I1) RAD51 - 31; (Job.3.S4.I2) RAD51 - 22; (Job.3.S4.I3) RAD51 - 31; (Job.3.S4.I4) RAD51 - 31 : **31.0**

**Job # 4**

(Job.4.S1.I1) RAD51 - 35; (Job.4.S1.I2) RAD51 - 37; (Job.4.S1.I3) RAD51 - 35; (Job.4.S1.I4) RAD51 - 37 : **36.0** (Job.4.S2.I1) RAD51 - 93; (Job.4.S2.I2) RAD51 - 95; (Job.4.S2.I3) RAD51 - 95; (Job.4.S2.I4) RAD51 - 93 : **94.0** (Job.4.S3.I1) RAD51 - 94; (Job.4.S3.I2) RAD51 - 109; (Job.4.S3.I3) RAD51 - 114; (Job.4.S3.I4) RAD51 - 116 : **111.5** (Job.4.S4.I1) RAD51 - 34; (Job.4.S4.I2) RAD51 - 34; (Job.4.S4.I3)

RAD51 - 34; (Job.4.S4.I4) RAD51 - 34 : **34.0**

*Best ranks of each set–island*: 12 - 25 - 142 - 32 - 5 - 28 - 28 - 29 - 29 - 36 - 34 - 6 - 29 - 25 - 25 - 22 - 35 - 93 - 94 - 34 -
*Median*: 34.0
*Median of best*: 29.0

### A.1.13 RB1CC1

**Job # 0**
(Job.0.S1.I1) RB1CC1 - 53; (Job.0.S1.I2) RB1CC1 - 52; (Job.0.S1.I3) RB1CC1 - 55; (Job.0.S1.I4) RB1CC1 - 54 : **53.5** (Job.0.S2.I1) RB1CC1 - 31; (Job.0.S2.I2) RB1CC1 - 25; (Job.0.S2.I3) RB1CC1 - 29; (Job.0.S2.I4) RB1CC1 - 31 : **30.0** (Job.0.S3.I1) RB1CC1 - 42; (Job.0.S3.I2) RB1CC1 - 44; (Job.0.S3.I3) RB1CC1 - 43; (Job.0.S3.I4) RB1CC1 - 44 : **43.5** (Job.0.S4.I1) RB1CC1 - 59; (Job.0.S4.I2) RB1CC1 - 59; (Job.0.S4.I3) RB1CC1 - 56; (Job.0.S4.I4) RB1CC1 - 62 : **59.0**

**Job # 1**
(Job.1.S1.I1) RB1CC1 - 23; (Job.1.S1.I2) RB1CC1 - 23; (Job.1.S1.I3) RB1CC1 - 23; (Job.1.S1.I4) RB1CC1 - 23 : **23.0** (Job.1.S2.I1) RB1CC1 - 1790; (Job.1.S2.I2) RB1CC1 - 1779; (Job.1.S2.I3) RB1CC1 - 1790; (Job.1.S2.I4) RB1CC1 - 1791 : **1790.0** (Job.1.S3.I1) RB1CC1 - 40; (Job.1.S3.I2) RB1CC1 - 43; (Job.1.S3.I3) RB1CC1 - 39; (Job.1.S3.I4) RB1CC1 - 43 : **41.5** (Job.1.S4.I1) RB1CC1 - 14; (Job.1.S4.I2) RB1CC1 - 14; (Job.1.S4.I3) RB1CC1 - 34; (Job.1.S4.I4) RB1CC1 - 20 : **17.0**

**Job # 2**
(Job.2.S1.I1) RB1CC1 - 62; (Job.2.S1.I2) RB1CC1 - 58; (Job.2.S1.I3) RB1CC1 - 53; (Job.2.S1.I4) RB1CC1 - 58 : **58.0** (Job.2.S2.I1) RB1CC1 - 9; (Job.2.S2.I2) RB1CC1 - 6; (Job.2.S2.I3) RB1CC1 - 6; (Job.2.S2.I4) RB1CC1 - 5 : **6.0** (Job.2.S3.I1) RB1CC1 - 46; (Job.2.S3.I2) RB1CC1 - 49; (Job.2.S3.I3) RB1CC1 - 48; (Job.2.S3.I4) RB1CC1 - 42 : **47.0** (Job.2.S4.I1) RB1CC1 - 40; (Job.2.S4.I2) RB1CC1 - 35; (Job.2.S4.I3) RB1CC1 - 38; (Job.2.S4.I4) RB1CC1 - 37 : **37.5**

**Job # 3**
(Job.3.S1.I1) RB1CC1 - 62; (Job.3.S1.I2) RB1CC1 - 62; (Job.3.S1.I3) RB1CC1 - 63; (Job.3.S1.I4) RB1CC1 - 61 : **62.0** (Job.3.S2.I1) RB1CC1 - 59; (Job.3.S2.I2) RB1CC1

- 59; (Job.3.S2.I3) RB1CC1 - 59; (Job.3.S2.I4) RB1CC1 - 59 : **59.0** (Job.3.S3.I1) RB1CC1 - 50; (Job.3.S3.I2) RB1CC1 - 53; (Job.3.S3.I3) RB1CC1 - 55; (Job.3.S3.I4) RB1CC1 - 50 : **51.5** (Job.3.S4.I1) RB1CC1 - 22; (Job.3.S4.I2) RB1CC1 - 23; (Job.3.S4.I3) RB1CC1 - 28; (Job.3.S4.I4) RB1CC1 - 27 : **25.0**

**Job # 4**
(Job.4.S1.I1) RB1CC1 - 59; (Job.4.S1.I2) RB1CC1 - 59; (Job.4.S1.I3) RB1CC1 - 59; (Job.4.S1.I4) RB1CC1 - 59 : **59.0** (Job.4.S2.I1) RB1CC1 - 53; (Job.4.S2.I2) RB1CC1 - 53; (Job.4.S2.I3) RB1CC1 - 54; (Job.4.S2.I4) RB1CC1 - 50 : **53.0** (Job.4.S3.I1) RB1CC1 - 32; (Job.4.S3.I2) RB1CC1 - 45; (Job.4.S3.I3) RB1CC1 - 155; (Job.4.S3.I4) RB1CC1 - 30 : **38.5** (Job.4.S4.I1) RB1CC1 - 40; (Job.4.S4.I2) RB1CC1 - 41; (Job.4.S4.I3) RB1CC1 - 48; (Job.4.S4.I4) RB1CC1 - 40 : **40.5**

*Best ranks of each set–island*: 52 - 25 - 42 - 56 - 23 - 1779 - 39 - 14 - 53 - 5 - 42 - 35 - 61 - 59 - 50 - 22 - 59 - 50 - 30 - 40 -
*Median*: 45.25
*Median of best*: 42.0

### A.1.14 STK11

**Job # 0**
(Job.0.S1.I1) STK11 - 497; (Job.0.S1.I2) STK11 - 484; (Job.0.S1.I3) STK11 - 605; (Job.0.S1.I4) STK11 - 730 : **551.0** (Job.0.S2.I1) STK11 - 602; (Job.0.S2.I2) STK11 - 796; (Job.0.S2.I3) STK11 - 583; (Job.0.S2.I4) STK11 - 566 : **592.5** (Job.0.S3.I1) STK11 - 1693; (Job.0.S3.I2) STK11 - 1595; (Job.0.S3.I3) STK11 - 1718; (Job.0.S3.I4) STK11 - 784 : **1644.0** (Job.0.S4.I1) STK11 - 357; (Job.0.S4.I2) STK11 - 358; (Job.0.S4.I3) STK11 - 374; (Job.0.S4.I4) STK11 - 176 : **357.5**

**Job # 1**
(Job.1.S1.I1) STK11 - 317; (Job.1.S1.I2) STK11 - 399; (Job.1.S1.I3) STK11 - 485; (Job.1.S1.I4) STK11 - 537 : **442.0** (Job.1.S2.I1) STK11 - 324; (Job.1.S2.I2) STK11 - 336; (Job.1.S2.I3) STK11 - 228; (Job.1.S2.I4) STK11 - 336 : **330.0** (Job.1.S3.I1) STK11 - 786; (Job.1.S3.I2) STK11 - 979; (Job.1.S3.I3) STK11 - 814; (Job.1.S3.I4) STK11 - 1626 : **896.5** (Job.1.S4.I1) STK11 - 152; (Job.1.S4.I2) STK11 - 151; (Job.1.S4.I3) STK11 - 151; (Job.1.S4.I4) STK11 - 612 : **151.5**

**Job # 2**

(Job.2.S1.I1) STK11 - 1587; (Job.2.S1.I2) STK11 - 1587; (Job.2.S1.I3) STK11 - 1587; (Job.2.S1.I4) STK11 - 1587 : **1587.0** (Job.2.S2.I1) STK11 - 454; (Job.2.S2.I2) STK11 - 585; (Job.2.S2.I3) STK11 - 592; (Job.2.S2.I4) STK11 - 594 : **588.5** (Job.2.S3.I1) STK11 - 152; (Job.2.S3.I2) STK11 - 152; (Job.2.S3.I3) STK11 - 151; (Job.2.S3.I4) STK11 - 160 : **152.0** (Job.2.S4.I1) STK11 - 603; (Job.2.S4.I2) STK11 - 605; (Job.2.S4.I3) STK11 - 159; (Job.2.S4.I4) STK11 - 556 : **579.5**

**Job # 3**

(Job.3.S1.I1) STK11 - 135; (Job.3.S1.I2) STK11 - 169; (Job.3.S1.I3) STK11 - 131; (Job.3.S1.I4) STK11 - 133 : **134.0** (Job.3.S2.I1) STK11 - 190; (Job.3.S2.I2) STK11 - 171; (Job.3.S2.I3) STK11 - 243; (Job.3.S2.I4) STK11 - 182 : **186.0** (Job.3.S3.I1) STK11 - 195; (Job.3.S3.I2) STK11 - 195; (Job.3.S3.I3) STK11 - 215; (Job.3.S3.I4) STK11 - 200 : **197.5** (Job.3.S4.I1) STK11 - 171; (Job.3.S4.I2) STK11 - 168; (Job.3.S4.I3) STK11 - 171; (Job.3.S4.I4) STK11 - 176 : **171.0**

**Job # 4**

(Job.4.S1.I1) STK11 - 221; (Job.4.S1.I2) STK11 - 128; (Job.4.S1.I3) STK11 - 221; (Job.4.S1.I4) STK11 - 116 : **174.5** (Job.4.S2.I1) STK11 - 29; (Job.4.S2.I2) STK11 - 27; (Job.4.S2.I3) STK11 - 32; (Job.4.S2.I4) STK11 - 29 : **29.0** (Job.4.S3.I1) STK11 - 394; (Job.4.S3.I2) STK11 - 403; (Job.4.S3.I3) STK11 - 200; (Job.4.S3.I4) STK11 - 179 : **297.0** (Job.4.S4.I1) STK11 - 1366; (Job.4.S4.I2) STK11 - 1371; (Job.4.S4.I3) STK11 - 1369; (Job.4.S4.I4) STK11 - 1304 : **1367.5**

*Best ranks of each set–island*: 484 - 566 - 784 - 176 - 317 - 228 - 786 - 151 - 1587 - 454 - 151 - 159 - 131 - 171 - 195 - 168 - 116 - 27 - 179 - 1304 -
*Median*: 343.75
*Median of best*: 187.0

### A.1.15 TP53

**Job # 0**

(Job.0.S1.I1) TP53 - 3; (Job.0.S1.I2) TP53 - 2; (Job.0.S1.I3) TP53 - 3; (Job.0.S1.I4) TP53 - 4 : **3.0** (Job.0.S2.I1) TP53 - 6; (Job.0.S2.I2) TP53 - 33; (Job.0.S2.I3) TP53 - 6; (Job.0.S2.I4) TP53 - 45 : **19.5** (Job.0.S3.I1) TP53 - 4; (Job.0.S3.I2) TP53 - 3; (Job.0.S3.I3) TP53 - 4; (Job.0.S3.I4) TP53 - 4 : **4.0** (Job.0.S4.I1) TP53 - 3;

(Job.0.S4.I2) TP53 - 5; (Job.0.S4.I3) TP53 - 3; (Job.0.S4.I4) TP53 - 2 : **3.0**

**Job # 1**

(Job.1.S1.I1) TP53 - 11; (Job.1.S1.I2) TP53 - 14; (Job.1.S1.I3) TP53 - 15; (Job.1.S1.I4) TP53 - 2 : **12.5** (Job.1.S2.I1) TP53 - 5; (Job.1.S2.I2) TP53 - 4; (Job.1.S2.I3) TP53 - 4; (Job.1.S2.I4) TP53 - 6 : **4.5** (Job.1.S3.I1) TP53 - 9; (Job.1.S3.I2) TP53 - 3; (Job.1.S3.I3) TP53 - 9; (Job.1.S3.I4) TP53 - 8 : **8.5** (Job.1.S4.I1) TP53 - 5; (Job.1.S4.I2) TP53 - 2; (Job.1.S4.I3) TP53 - 5; (Job.1.S4.I4) TP53 - 3 : **4.0**

**Job # 2**

(Job.2.S1.I1) TP53 - 28; (Job.2.S1.I2) TP53 - 28; (Job.2.S1.I3) TP53 - 3; (Job.2.S1.I4) TP53 - 3 : **15.5** (Job.2.S2.I1) TP53 - 1; (Job.2.S2.I2) TP53 - 3; (Job.2.S2.I3) TP53 - 3; (Job.2.S2.I4) TP53 - 1 : **2.0** (Job.2.S3.I1) TP53 - 2; (Job.2.S3.I2) TP53 - 4; (Job.2.S3.I3) TP53 - 4; (Job.2.S3.I4) TP53 - 1 : **3.0** (Job.2.S4.I1) TP53 - 5; (Job.2.S4.I2) TP53 - 5; (Job.2.S4.I3) TP53 - 5; (Job.2.S4.I4) TP53 - 5 : **5.0**

**Job # 3**

(Job.3.S1.I1) TP53 - 7; (Job.3.S1.I2) TP53 - 7; (Job.3.S1.I3) TP53 - 6; (Job.3.S1.I4) TP53 - 8 : **7.0** (Job.3.S2.I1) TP53 - 2; (Job.3.S2.I2) TP53 - 2; (Job.3.S2.I3) TP53 - 15; (Job.3.S2.I4) TP53 - 2 : **2.0** (Job.3.S3.I1) TP53 - 3; (Job.3.S3.I2) TP53 - 3; (Job.3.S3.I3) TP53 - 3; (Job.3.S3.I4) TP53 - 3 : **3.0** (Job.3.S4.I1) TP53 - 2; (Job.3.S4.I2) TP53 - 2; (Job.3.S4.I3) TP53 - 2; (Job.3.S4.I4) TP53 - 2 : **2.0**

**Job # 4**

(Job.4.S1.I1) TP53 - 5; (Job.4.S1.I2) TP53 - 5; (Job.4.S1.I3) TP53 - 5; (Job.4.S1.I4) TP53 - 5 : **5.0** (Job.4.S2.I1) TP53 - 4; (Job.4.S2.I2) TP53 - 4; (Job.4.S2.I3) TP53 - 4; (Job.4.S2.I4) TP53 - 4 : **4.0** (Job.4.S3.I1) TP53 - 2; (Job.4.S3.I2) TP53 - 2; (Job.4.S3.I3) TP53 - 2; (Job.4.S3.I4) TP53 - 2 : **2.0** (Job.4.S4.I1) TP53 - 3; (Job.4.S4.I2) TP53 - 3; (Job.4.S4.I3) TP53 - 4; (Job.4.S4.I4) TP53 - 2 : **3.0**

*Best ranks of each set–island*: 2 - 6 - 3 - 2 - 2 - 4 - 3 - 2 - 3 - 1 - 1 - 5 - 6 - 2 - 3 - 2 - 5 - 4 - 2 - 2 -
*Median*: 4.0
*Median of best*: 2.5

## A.2 Parkinson's Disease

### A.2.1 APOE

**Job # 0**

(Job.0.S1.I1) APOE - 285; (Job.0.S1.I2) APOE - 285; (Job.0.S1.I3) APOE - 285; (Job.0.S1.I4) APOE - 63 : **285.0** (Job.0.S2.I1) APOE - 43; (Job.0.S2.I2) APOE - 43; (Job.0.S2.I3) APOE - 46; (Job.0.S2.I4) APOE - 43 : **43.0** (Job.0.S3.I1) APOE - 105; (Job.0.S3.I2) APOE - 98; (Job.0.S3.I3) APOE - 120; (Job.0.S3.I4) APOE - 105 : **105.0** (Job.0.S4.I1) APOE - 287; (Job.0.S4.I2) APOE - 213; (Job.0.S4.I3) APOE - 253; (Job.0.S4.I4) APOE - 293 : **270.0**

**Job # 1**

(Job.1.S1.I1) APOE - 29; (Job.1.S1.I2) APOE - 8; (Job.1.S1.I3) APOE - 124; (Job.1.S1.I4) APOE - 15 : **22.0** (Job.1.S2.I1) APOE - 10; (Job.1.S2.I2) APOE - 57; (Job.1.S2.I3) APOE - 24; (Job.1.S2.I4) APOE - 34 : **29.0** (Job.1.S3.I1) APOE - 97; (Job.1.S3.I2) APOE - 400; (Job.1.S3.I3) APOE - 156; (Job.1.S3.I4) APOE - 400 : **278.0** (Job.1.S4.I1) APOE - 173; (Job.1.S4.I2) APOE - 173; (Job.1.S4.I3) APOE - 173; (Job.1.S4.I4) APOE - 173 : **173.0**

**Job # 2**

(Job.2.S1.I1) APOE - 178; (Job.2.S1.I2) APOE - 171; (Job.2.S1.I3) APOE - 72; (Job.2.S1.I4) APOE - 170 : **170.5** (Job.2.S2.I1) APOE - 102; (Job.2.S2.I2) APOE - 115; (Job.2.S2.I3) APOE - 102; (Job.2.S2.I4) APOE - 109 : **105.5** (Job.2.S3.I1) APOE - 83; (Job.2.S3.I2) APOE - 183; (Job.2.S3.I3) APOE - 80; (Job.2.S3.I4) APOE - 82 : **82.5** (Job.2.S4.I1) APOE - 63; (Job.2.S4.I2) APOE - 61; (Job.2.S4.I3) APOE - 73; (Job.2.S4.I4) APOE - 67 : **65.0**

**Job # 3**

(Job.3.S1.I1) APOE - 67; (Job.3.S1.I2) APOE - 65; (Job.3.S1.I3) APOE - 66; (Job.3.S1.I4) APOE - 67 : **66.5** (Job.3.S2.I1) APOE - 18; (Job.3.S2.I2) APOE - 9; (Job.3.S2.I3) APOE - 18; (Job.3.S2.I4) APOE - 8 : **13.5** (Job.3.S3.I1) APOE - 12; (Job.3.S3.I2) APOE - 20; (Job.3.S3.I3) APOE - 12; (Job.3.S3.I4) APOE - 20 : **16.0** (Job.3.S4.I1) APOE - 108; (Job.3.S4.I2) APOE - 110; (Job.3.S4.I3) APOE - 110; (Job.3.S4.I4) APOE - 108 : **109.0**

**Job # 4**

(Job.4.S1.I1) APOE - 585; (Job.4.S1.I2) APOE - 603; (Job.4.S1.I3) APOE - 587; (Job.4.S1.I4) APOE - 566 : **586.0** (Job.4.S2.I1) APOE - 26; (Job.4.S2.I2) APOE - 32; (Job.4.S2.I3) APOE - 26; (Job.4.S2.I4) APOE - 72 : **29.0** (Job.4.S3.I1) APOE - 35; (Job.4.S3.I2) APOE - 38; (Job.4.S3.I3) APOE - 131; (Job.4.S3.I4) APOE - 38 : **38.0** (Job.4.S4.I1) APOE - 135; (Job.4.S4.I2) APOE - 98; (Job.4.S4.I3) APOE - 101; (Job.4.S4.I4) APOE - 135 : **118.0**

*Best ranks of each set–island*: 63 - 43 - 98 - 213 - 8 - 10 - 97 - 173 - 72 - 102 - 80 - 61 - 65 - 8 - 12 - 108 - 566 - 26 - 35 - 98 -
*Median*: 93.75
*Median of best*: 68.5

### A.2.2 BDNF

**Job # 0**
(Job.0.S1.I1) BDNF - 502; (Job.0.S1.I2) BDNF - 479; (Job.0.S1.I3) BDNF - 474; (Job.0.S1.I4) BDNF - 462 : **476.5** (Job.0.S2.I1) BDNF - 116; (Job.0.S2.I2) BDNF - 220; (Job.0.S2.I3) BDNF - 112; (Job.0.S2.I4) BDNF - 225 : **168.0** (Job.0.S3.I1) BDNF - 1003; (Job.0.S3.I2) BDNF - 695; (Job.0.S3.I3) BDNF - 960; (Job.0.S3.I4) BDNF - 879 : **919.5** (Job.0.S4.I1) BDNF - 12; (Job.0.S4.I2) BDNF - 14; (Job.0.S4.I3) BDNF - 52; (Job.0.S4.I4) BDNF - 14 : **14.0**

**Job # 1**
(Job.1.S1.I1) BDNF - 189; (Job.1.S1.I2) BDNF - 190; (Job.1.S1.I3) BDNF - 189; (Job.1.S1.I4) BDNF - 189 : **189.0** (Job.1.S2.I1) BDNF - 2692; (Job.1.S2.I2) BDNF - 2693; (Job.1.S2.I3) BDNF - 2693; (Job.1.S2.I4) BDNF - 2694 : **2693.0** (Job.1.S3.I1) BDNF - 116; (Job.1.S3.I2) BDNF - 163; (Job.1.S3.I3) BDNF - 163; (Job.1.S3.I4) BDNF - 184 : **163.0** (Job.1.S4.I1) BDNF - 8; (Job.1.S4.I2) BDNF - 8; (Job.1.S4.I3) BDNF - 8; (Job.1.S4.I4) BDNF - 8 : **8.0**

**Job # 2**
(Job.2.S1.I1) BDNF - 800; (Job.2.S1.I2) BDNF - 12; (Job.2.S1.I3) BDNF - 747; (Job.2.S1.I4) BDNF - 519 : **633.0** (Job.2.S2.I1) BDNF - 40; (Job.2.S2.I2) BDNF - 40; (Job.2.S2.I3) BDNF - 40; (Job.2.S2.I4) BDNF - 40 : **40.0** (Job.2.S3.I1) BDNF - 150; (Job.2.S3.I2) BDNF - 149; (Job.2.S3.I3) BDNF - 150; (Job.2.S3.I4) BDNF - 149 : **149.5** (Job.2.S4.I1) BDNF - 198; (Job.2.S4.I2) BDNF - 194; (Job.2.S4.I3) BDNF -

200; (Job.2.S4.I4) BDNF - 149 : **196.0**

**Job # 3**

(Job.3.S1.I1) BDNF - 15; (Job.3.S1.I2) BDNF - 55; (Job.3.S1.I3) BDNF - 27; (Job.3.S1.I4) BDNF - 15 : **21.0** (Job.3.S2.I1) BDNF - 59; (Job.3.S2.I2) BDNF - 59; (Job.3.S2.I3) BDNF - 51; (Job.3.S2.I4) BDNF - 61 : **59.0** (Job.3.S3.I1) BDNF - 941; (Job.3.S3.I2) BDNF - 859; (Job.3.S3.I3) BDNF - 930; (Job.3.S3.I4) BDNF - 936 : **933.0** (Job.3.S4.I1) BDNF - 111; (Job.3.S4.I2) BDNF - 111; (Job.3.S4.I3) BDNF - 111; (Job.3.S4.I4) BDNF - 111 : **111.0**

**Job # 4**

(Job.4.S1.I1) BDNF - 72; (Job.4.S1.I2) BDNF - 72; (Job.4.S1.I3) BDNF - 72; (Job.4.S1.I4) BDNF - 57 : **72.0** (Job.4.S2.I1) BDNF - 56; (Job.4.S2.I2) BDNF - 109; (Job.4.S2.I3) BDNF - 112; (Job.4.S2.I4) BDNF - 118 : **110.5** (Job.4.S3.I1) BDNF - 7; (Job.4.S3.I2) BDNF - 7; (Job.4.S3.I3) BDNF - 99; (Job.4.S3.I4) BDNF - 7 : **7.0** (Job.4.S4.I1) BDNF - 213; (Job.4.S4.I2) BDNF - 213; (Job.4.S4.I3) BDNF - 213; (Job.4.S4.I4) BDNF - 213 : **213.0**

*Best ranks of each set–island*: 462 - 112 - 695 - 12 - 189 - 2692 - 116 - 8 - 12 - 40 - 149 - 149 - 15 - 51 - 859 - 111 - 57 - 56 - 7 - 213 -
*Median*: 156.25
*Median of best*: 111.5

### A.2.3 BST1

**Job # 0**

(Job.0.S1.I1) BST1 - 10; (Job.0.S1.I2) BST1 - 10; (Job.0.S1.I3) BST1 - 55; (Job.0.S1.I4) BST1 - 108 : **32.5** (Job.0.S2.I1) BST1 - 7; (Job.0.S2.I2) BST1 - 6; (Job.0.S2.I3) BST1 - 5; (Job.0.S2.I4) BST1 - 5 : **5.5** (Job.0.S3.I1) BST1 - 44; (Job.0.S3.I2) BST1 - 64; (Job.0.S3.I3) BST1 - 37; (Job.0.S3.I4) BST1 - 35 : **40.5** (Job.0.S4.I1) BST1 - 4; (Job.0.S4.I2) BST1 - 4; (Job.0.S4.I3) BST1 - 4; (Job.0.S4.I4) BST1 - 4 : **4.0**

**Job # 1**

(Job.1.S1.I1) BST1 - 5; (Job.1.S1.I2) BST1 - 5; (Job.1.S1.I3) BST1 - 5; (Job.1.S1.I4) BST1 - 5 : **5.0** (Job.1.S2.I1) BST1 - 31; (Job.1.S2.I2) BST1 - 31; (Job.1.S2.I3) BST1 - 34; (Job.1.S2.I4) BST1 - 34 : **32.5** (Job.1.S3.I1) BST1 - 3; (Job.1.S3.I2) BST1 -

13; (Job.1.S3.I3) BST1 - 5; (Job.1.S3.I4) BST1 - 6 : **5.5** (Job.1.S4.I1) BST1 - 17; (Job.1.S4.I2) BST1 - 5; (Job.1.S4.I3) BST1 - 5; (Job.1.S4.I4) BST1 - 5 : **5.0**

**Job # 2**

(Job.2.S1.I1) BST1 - 6; (Job.2.S1.I2) BST1 - 6; (Job.2.S1.I3) BST1 - 6; (Job.2.S1.I4) BST1 - 5 : **6.0** (Job.2.S2.I1) BST1 - 19; (Job.2.S2.I2) BST1 - 13; (Job.2.S2.I3) BST1 - 22; (Job.2.S2.I4) BST1 - 36 : **20.5** (Job.2.S3.I1) BST1 - 4; (Job.2.S3.I2) BST1 - 4; (Job.2.S3.I3) BST1 - 4; (Job.2.S3.I4) BST1 - 4 : **4.0** (Job.2.S4.I1) BST1 - 14; (Job.2.S4.I2) BST1 - 14; (Job.2.S4.I3) BST1 - 12; (Job.2.S4.I4) BST1 - 13 : **13.5**

**Job # 3**

(Job.3.S1.I1) BST1 - 38; (Job.3.S1.I2) BST1 - 19; (Job.3.S1.I3) BST1 - 27; (Job.3.S1.I4) BST1 - 19 : **23.0** (Job.3.S2.I1) BST1 - 18; (Job.3.S2.I2) BST1 - 5; (Job.3.S2.I3) BST1 - 5; (Job.3.S2.I4) BST1 - 5 : **5.0** (Job.3.S3.I1) BST1 - 5; (Job.3.S3.I2) BST1 - 5; (Job.3.S3.I3) BST1 - 5; (Job.3.S3.I4) BST1 - 5 : **5.0** (Job.3.S4.I1) BST1 - 5; (Job.3.S4.I2) BST1 - 5; (Job.3.S4.I3) BST1 - 5; (Job.3.S4.I4) BST1 - 12 : **5.0**

**Job # 4**

(Job.4.S1.I1) BST1 - 342; (Job.4.S1.I2) BST1 - 145; (Job.4.S1.I3) BST1 - 376; (Job.4.S1.I4) BST1 - 9 : **243.5** (Job.4.S2.I1) BST1 - 8; (Job.4.S2.I2) BST1 - 8; (Job.4.S2.I3) BST1 - 21; (Job.4.S2.I4) BST1 - 6 : **8.0** (Job.4.S3.I1) BST1 - 18; (Job.4.S3.I2) BST1 - 15; (Job.4.S3.I3) BST1 - 27; (Job.4.S3.I4) BST1 - 18 : **18.0** (Job.4.S4.I1) BST1 - 6; (Job.4.S4.I2) BST1 - 6; (Job.4.S4.I3) BST1 - 5; (Job.4.S4.I4) BST1 - 5 : **5.5**

*Best ranks of each set–island*: 10 - 5 - 35 - 4 - 5 - 31 - 3 - 5 - 5 - 13 - 4 - 12 - 19 - 5 - 5 - 5 - 9 - 6 - 15 - 5 -
*Median*: 5.75
*Median of best*: 5.0

### A.2.4 COMT

**Job # 0**

(Job.0.S1.I1) COMT - 756; (Job.0.S1.I2) COMT - 753; (Job.0.S1.I3) COMT - 771; (Job.0.S1.I4) COMT - 749 : **754.5** (Job.0.S2.I1) COMT - 47; (Job.0.S2.I2) COMT - 23; (Job.0.S2.I3) COMT - 11; (Job.0.S2.I4) COMT - 60 : **35.0** (Job.0.S3.I1) COMT - 22; (Job.0.S3.I2) COMT - 48; (Job.0.S3.I3) COMT - 612; (Job.0.S3.I4) COMT - 25

: **36.5** (Job.0.S4.I1) COMT - 698; (Job.0.S4.I2) COMT - 698; (Job.0.S4.I3) COMT - 698; (Job.0.S4.I4) COMT - 698 : **698.0**

**Job # 1**

(Job.1.S1.I1) COMT - 656; (Job.1.S1.I2) COMT - 876; (Job.1.S1.I3) COMT - 874; (Job.1.S1.I4) COMT - 656 : **765.0** (Job.1.S2.I1) COMT - 270; (Job.1.S2.I2) COMT - 286; (Job.1.S2.I3) COMT - 273; (Job.1.S2.I4) COMT - 295 : **279.5** (Job.1.S3.I1) COMT - 362; (Job.1.S3.I2) COMT - 358; (Job.1.S3.I3) COMT - 341; (Job.1.S3.I4) COMT - 372 : **360.0** (Job.1.S4.I1) COMT - 204; (Job.1.S4.I2) COMT - 205; (Job.1.S4.I3) COMT - 230; (Job.1.S4.I4) COMT - 187 : **204.5**

**Job # 2**

(Job.2.S1.I1) COMT - 240; (Job.2.S1.I2) COMT - 240; (Job.2.S1.I3) COMT - 99; (Job.2.S1.I4) COMT - 123 : **181.5** (Job.2.S2.I1) COMT - 144; (Job.2.S2.I2) COMT - 150; (Job.2.S2.I3) COMT - 151; (Job.2.S2.I4) COMT - 489 : **150.5** (Job.2.S3.I1) COMT - 10; (Job.2.S3.I2) COMT - 13; (Job.2.S3.I3) COMT - 10; (Job.2.S3.I4) COMT - 41 : **11.5** (Job.2.S4.I1) COMT - 143; (Job.2.S4.I2) COMT - 143; (Job.2.S4.I3) COMT - 148; (Job.2.S4.I4) COMT - 149 : **145.5**

**Job # 3**

(Job.3.S1.I1) COMT - 261; (Job.3.S1.I2) COMT - 393; (Job.3.S1.I3) COMT - 23; (Job.3.S1.I4) COMT - 12 : **142.0** (Job.3.S2.I1) COMT - 2452; (Job.3.S2.I2) COMT - 2452; (Job.3.S2.I3) COMT - 2451; (Job.3.S2.I4) COMT - 2450 : **2451.5** (Job.3.S3.I1) COMT - 951; (Job.3.S3.I2) COMT - 931; (Job.3.S3.I3) COMT - 947; (Job.3.S3.I4) COMT - 951 : **949.0** (Job.3.S4.I1) COMT - 248; (Job.3.S4.I2) COMT - 215; (Job.3.S4.I3) COMT - 295; (Job.3.S4.I4) COMT - 213 : **231.5**

**Job # 4**

(Job.4.S1.I1) COMT - 214; (Job.4.S1.I2) COMT - 264; (Job.4.S1.I3) COMT - 256; (Job.4.S1.I4) COMT - 426 : **260.0** (Job.4.S2.I1) COMT - 377; (Job.4.S2.I2) COMT - 378; (Job.4.S2.I3) COMT - 417; (Job.4.S2.I4) COMT - 340 : **377.5** (Job.4.S3.I1) COMT - 1000; (Job.4.S3.I2) COMT - 914; (Job.4.S3.I3) COMT - 918; (Job.4.S3.I4) COMT - 146 : **916.0** (Job.4.S4.I1) COMT - 209; (Job.4.S4.I2) COMT - 177; (Job.4.S4.I3) COMT - 187; (Job.4.S4.I4) COMT - 183 : **185.0**

*Best ranks of each set–island*: 749 - 11 - 22 - 698 - 656 - 270 - 341 - 187 - 99 -

144 - 10 - 143 - 12 - 2450 - 931 - 213 - 214 - 340 - 146 - 177 -
*Median*: 245.75
*Median of best*: 200.0

### A.2.5 CYP2D6

**Job # 0**
(Job.0.S1.I1) CYP2D6 - 564; (Job.0.S1.I2) CYP2D6 - 727; (Job.0.S1.I3) CYP2D6 - 846; (Job.0.S1.I4) CYP2D6 - 845 : **786.0** (Job.0.S2.I1) CYP2D6 - 854; (Job.0.S2.I2) CYP2D6 - 892; (Job.0.S2.I3) CYP2D6 - 885; (Job.0.S2.I4) CYP2D6 - 915 : **888.5** (Job.0.S3.I1) CYP2D6 - 137; (Job.0.S3.I2) CYP2D6 - 791; (Job.0.S3.I3) CYP2D6 - 779; (Job.0.S3.I4) CYP2D6 - 190 : **484.5** (Job.0.S4.I1) CYP2D6 - 766; (Job.0.S4.I2) CYP2D6 - 764; (Job.0.S4.I3) CYP2D6 - 568; (Job.0.S4.I4) CYP2D6 - 576 : **670.0**

**Job # 1**
(Job.1.S1.I1) CYP2D6 - 940; (Job.1.S1.I2) CYP2D6 - 949; (Job.1.S1.I3) CYP2D6 - 951; (Job.1.S1.I4) CYP2D6 - 929 : **944.5** (Job.1.S2.I1) CYP2D6 - 947; (Job.1.S2.I2) CYP2D6 - 1501; (Job.1.S2.I3) CYP2D6 - 962; (Job.1.S2.I4) CYP2D6 - 944 : **954.5** (Job.1.S3.I1) CYP2D6 - 166; (Job.1.S3.I2) CYP2D6 - 162; (Job.1.S3.I3) CYP2D6 - 271; (Job.1.S3.I4) CYP2D6 - 170 : **168.0** (Job.1.S4.I1) CYP2D6 - 733; (Job.1.S4.I2) CYP2D6 - 739; (Job.1.S4.I3) CYP2D6 - 979; (Job.1.S4.I4) CYP2D6 - 733 : **736.0**

**Job # 2**
(Job.2.S1.I1) CYP2D6 - 253; (Job.2.S1.I2) CYP2D6 - 187; (Job.2.S1.I3) CYP2D6 - 277; (Job.2.S1.I4) CYP2D6 - 268 : **260.5** (Job.2.S2.I1) CYP2D6 - 1009; (Job.2.S2.I2) CYP2D6 - 816; (Job.2.S2.I3) CYP2D6 - 1079; (Job.2.S2.I4) CYP2D6 - 1099 : **1044.0** (Job.2.S3.I1) CYP2D6 - 962; (Job.2.S3.I2) CYP2D6 - 825; (Job.2.S3.I3) CYP2D6 - 828; (Job.2.S3.I4) CYP2D6 - 943 : **885.5** (Job.2.S4.I1) CYP2D6 - 1961; (Job.2.S4.I2) CYP2D6 - 1962; (Job.2.S4.I3) CYP2D6 - 1905; (Job.2.S4.I4) CYP2D6 - 1938 : **1949.5**

**Job # 3**
(Job.3.S1.I1) CYP2D6 - 369; (Job.3.S1.I2) CYP2D6 - 369; (Job.3.S1.I3) CYP2D6 - 1361; (Job.3.S1.I4) CYP2D6 - 1704 : **865.0** (Job.3.S2.I1) CYP2D6 - 768; (Job.3.S2.I2) CYP2D6 - 718; (Job.3.S2.I3) CYP2D6 - 744; (Job.3.S2.I4) CYP2D6 - 649 : **731.0** (Job.3.S3.I1) CYP2D6 - 952; (Job.3.S3.I2) CYP2D6 - 653; (Job.3.S3.I3) CYP2D6 - 957; (Job.3.S3.I4) CYP2D6 - 1503 : **954.5** (Job.3.S4.I1) CYP2D6 - 405; (Job.3.S4.I2)

CYP2D6 - 405; (Job.3.S4.I3) CYP2D6 - 405; (Job.3.S4.I4) CYP2D6 - 405 : **405.0**

**Job # 4**

(Job.4.S1.I1) CYP2D6 - 822; (Job.4.S1.I2) CYP2D6 - 1967; (Job.4.S1.I3) CYP2D6 - 1224; (Job.4.S1.I4) CYP2D6 - 1125 : **1174.5** (Job.4.S2.I1) CYP2D6 - 890; (Job.4.S2.I2) CYP2D6 - 926; (Job.4.S2.I3) CYP2D6 - 926; (Job.4.S2.I4) CYP2D6 - 938 : **926.0** (Job.4.S3.I1) CYP2D6 - 978; (Job.4.S3.I2) CYP2D6 - 990; (Job.4.S3.I3) CYP2D6 - 981; (Job.4.S3.I4) CYP2D6 - 978 : **979.5** (Job.4.S4.I1) CYP2D6 - 370; (Job.4.S4.I2) CYP2D6 - 540; (Job.4.S4.I3) CYP2D6 - 515; (Job.4.S4.I4) CYP2D6 - 382 : **448.5**

*Best ranks of each set–island*: 564 - 854 - 137 - 568 - 929 - 944 - 162 - 733 - 187 - 816 - 825 - 1905 - 369 - 649 - 653 - 405 - 822 - 890 - 978 - 370 -
*Median*: 875.25
*Median of best*: 693.0

### A.2.6 DRD2

**Job # 0**

(Job.0.S1.I1) DRD2 - 605; (Job.0.S1.I2) DRD2 - 431; (Job.0.S1.I3) DRD2 - 608; (Job.0.S1.I4) DRD2 - 609 : **606.5** (Job.0.S2.I1) DRD2 - 110; (Job.0.S2.I2) DRD2 - 108; (Job.0.S2.I3) DRD2 - 106; (Job.0.S2.I4) DRD2 - 104 : **107.0** (Job.0.S3.I1) DRD2 - 933; (Job.0.S3.I2) DRD2 - 931; (Job.0.S3.I3) DRD2 - 545; (Job.0.S3.I4) DRD2 - 933 : **932.0** (Job.0.S4.I1) DRD2 - 106; (Job.0.S4.I2) DRD2 - 109; (Job.0.S4.I3) DRD2 - 109; (Job.0.S4.I4) DRD2 - 106 : **107.5**

**Job # 1**

(Job.1.S1.I1) DRD2 - 850; (Job.1.S1.I2) DRD2 - 807; (Job.1.S1.I3) DRD2 - 162; (Job.1.S1.I4) DRD2 - 789 : **798.0** (Job.1.S2.I1) DRD2 - 201; (Job.1.S2.I2) DRD2 - 213; (Job.1.S2.I3) DRD2 - 210; (Job.1.S2.I4) DRD2 - 215 : **211.5** (Job.1.S3.I1) DRD2 - 864; (Job.1.S3.I2) DRD2 - 398; (Job.1.S3.I3) DRD2 - 863; (Job.1.S3.I4) DRD2 - 825 : **844.0** (Job.1.S4.I1) DRD2 - 962; (Job.1.S4.I2) DRD2 - 1021; (Job.1.S4.I3) DRD2 - 985; (Job.1.S4.I4) DRD2 - 894 : **973.5**

**Job # 2**

(Job.2.S1.I1) DRD2 - 1118; (Job.2.S1.I2) DRD2 - 113; (Job.2.S1.I3) DRD2 - 1115; (Job.2.S1.I4) DRD2 - 888 : **1001.5** (Job.2.S2.I1) DRD2 - 334; (Job.2.S2.I2) DRD2

- 325; (Job.2.S2.I3) DRD2 - 514; (Job.2.S2.I4) DRD2 - 324 : **329.5** (Job.2.S3.I1) DRD2 - 86; (Job.2.S3.I2) DRD2 - 86; (Job.2.S3.I3) DRD2 - 86; (Job.2.S3.I4) DRD2 - 86 : **86.0** (Job.2.S4.I1) DRD2 - 144; (Job.2.S4.I2) DRD2 - 144; (Job.2.S4.I3) DRD2 - 184; (Job.2.S4.I4) DRD2 - 190 : **164.0**

**Job # 3**
(Job.3.S1.I1) DRD2 - 947; (Job.3.S1.I2) DRD2 - 945; (Job.3.S1.I3) DRD2 - 969; (Job.3.S1.I4) DRD2 - 948 : **947.5** (Job.3.S2.I1) DRD2 - 107; (Job.3.S2.I2) DRD2 - 107; (Job.3.S2.I3) DRD2 - 107; (Job.3.S2.I4) DRD2 - 107 : **107.0** (Job.3.S3.I1) DRD2 - 199; (Job.3.S3.I2) DRD2 - 395; (Job.3.S3.I3) DRD2 - 144; (Job.3.S3.I4) DRD2 - 199 : **199.0** (Job.3.S4.I1) DRD2 - 545; (Job.3.S4.I2) DRD2 - 533; (Job.3.S4.I3) DRD2 - 543; (Job.3.S4.I4) DRD2 - 532 : **538.0**

**Job # 4**
(Job.4.S1.I1) DRD2 - 316; (Job.4.S1.I2) DRD2 - 203; (Job.4.S1.I3) DRD2 - 134; (Job.4.S1.I4) DRD2 - 134 : **168.5** (Job.4.S2.I1) DRD2 - 333; (Job.4.S2.I2) DRD2 - 331; (Job.4.S2.I3) DRD2 - 349; (Job.4.S2.I4) DRD2 - 347 : **340.0** (Job.4.S3.I1) DRD2 - 910; (Job.4.S3.I2) DRD2 - 798; (Job.4.S3.I3) DRD2 - 798; (Job.4.S3.I4) DRD2 - 912 : **854.0** (Job.4.S4.I1) DRD2 - 612; (Job.4.S4.I2) DRD2 - 564; (Job.4.S4.I3) DRD2 - 571; (Job.4.S4.I4) DRD2 - 564 : **567.5**

*Best ranks of each set–island*: 431 - 104 - 545 - 106 - 162 - 201 - 398 - 894 - 113 - 324 - 86 - 144 - 945 - 107 - 144 - 532 - 134 - 331 - 798 - 564 -
*Median*: 439.0
*Median of best*: 262.5

### A.2.7 GAK

**Job # 0**
(Job.0.S1.I1) GAK - 303; (Job.0.S1.I2) GAK - 335; (Job.0.S1.I3) GAK - 298; (Job.0.S1.I4) GAK - 245 : **300.5** (Job.0.S2.I1) GAK - 53; (Job.0.S2.I2) GAK - 122; (Job.0.S2.I3) GAK - 122; (Job.0.S2.I4) GAK - 385 : **122.0** (Job.0.S3.I1) GAK - 307; (Job.0.S3.I2) GAK - 230; (Job.0.S3.I3) GAK - 284; (Job.0.S3.I4) GAK - 241 : **262.5** (Job.0.S4.I1) GAK - 63; (Job.0.S4.I2) GAK - 46; (Job.0.S4.I3) GAK - 82; (Job.0.S4.I4) GAK - 60 : **61.5**

**Job # 1**

(Job.1.S1.I1) GAK - 588; (Job.1.S1.I2) GAK - 538; (Job.1.S1.I3) GAK - 591; (Job.1.S1.I4) GAK - 594 : **589.5** (Job.1.S2.I1) GAK - 153; (Job.1.S2.I2) GAK - 153; (Job.1.S2.I3) GAK - 160; (Job.1.S2.I4) GAK - 277 : **156.5** (Job.1.S3.I1) GAK - 191; (Job.1.S3.I2) GAK - 215; (Job.1.S3.I3) GAK - 222; (Job.1.S3.I4) GAK - 218 : **216.5** (Job.1.S4.I1) GAK - 319; (Job.1.S4.I2) GAK - 319; (Job.1.S4.I3) GAK - 319; (Job.1.S4.I4) GAK - 318 : **319.0**

**Job # 2**

(Job.2.S1.I1) GAK - 449; (Job.2.S1.I2) GAK - 425; (Job.2.S1.I3) GAK - 847; (Job.2.S1.I4) GAK - 847 : **648.0** (Job.2.S2.I1) GAK - 506; (Job.2.S2.I2) GAK - 506; (Job.2.S2.I3) GAK - 507; (Job.2.S2.I4) GAK - 506 : **506.0** (Job.2.S3.I1) GAK - 462; (Job.2.S3.I2) GAK - 262; (Job.2.S3.I3) GAK - 256; (Job.2.S3.I4) GAK - 256 : **259.0** (Job.2.S4.I1) GAK - 748; (Job.2.S4.I2) GAK - 748; (Job.2.S4.I3) GAK - 748; (Job.2.S4.I4) GAK - 876 : **748.0**

**Job # 3**

(Job.3.S1.I1) GAK - 837; (Job.3.S1.I2) GAK - 862; (Job.3.S1.I3) GAK - 862; (Job.3.S1.I4) GAK - 837 : **849.5** (Job.3.S2.I1) GAK - 935; (Job.3.S2.I2) GAK - 935; (Job.3.S2.I3) GAK - 935; (Job.3.S2.I4) GAK - 935 : **935.0** (Job.3.S3.I1) GAK - 561; (Job.3.S3.I2) GAK - 561; (Job.3.S3.I3) GAK - 558; (Job.3.S3.I4) GAK - 574 : **561.0** (Job.3.S4.I1) GAK - 213; (Job.3.S4.I2) GAK - 225; (Job.3.S4.I3) GAK - 225; (Job.3.S4.I4) GAK - 222 : **223.5**

**Job # 4**

(Job.4.S1.I1) GAK - 185; (Job.4.S1.I2) GAK - 188; (Job.4.S1.I3) GAK - 139; (Job.4.S1.I4) GAK - 153 : **169.0** (Job.4.S2.I1) GAK - 151; (Job.4.S2.I2) GAK - 113; (Job.4.S2.I3) GAK - 130; (Job.4.S2.I4) GAK - 133 : **131.5** (Job.4.S3.I1) GAK - 152; (Job.4.S3.I2) GAK - 150; (Job.4.S3.I3) GAK - 157; (Job.4.S3.I4) GAK - 148 : **151.0** (Job.4.S4.I1) GAK - 909; (Job.4.S4.I2) GAK - 808; (Job.4.S4.I3) GAK - 857; (Job.4.S4.I4) GAK - 709 : **832.5**

*Best ranks of each set–island*: 245 - 53 - 230 - 46 - 538 - 153 - 191 - 318 - 425 - 506 - 256 - 748 - 837 - 935 - 558 - 213 - 139 - 113 - 148 - 709 -
*Median*: 281.5
*Median of best*: 250.5

### A.2.8 GBA

**Job # 0**

(Job.0.S1.I1) GBA - 1716; (Job.0.S1.I2) GBA - 1961; (Job.0.S1.I3) GBA - 2379; (Job.0.S1.I4) GBA - 1660 : **1838.5** (Job.0.S2.I1) GBA - 1454; (Job.0.S2.I2) GBA - 1911; (Job.0.S2.I3) GBA - 1434; (Job.0.S2.I4) GBA - 1454 : **1454.0** (Job.0.S3.I1) GBA - 1754; (Job.0.S3.I2) GBA - 1759; (Job.0.S3.I3) GBA - 1759; (Job.0.S3.I4) GBA - 1823 : **1759.0** (Job.0.S4.I1) GBA - 1855; (Job.0.S4.I2) GBA - 1794; (Job.0.S4.I3) GBA - 1872; (Job.0.S4.I4) GBA - 1866 : **1860.5**

**Job # 1**

(Job.1.S1.I1) GBA - 1920; (Job.1.S1.I2) GBA - 1920; (Job.1.S1.I3) GBA - 1920; (Job.1.S1.I4) GBA - 1920 : **1920.0** (Job.1.S2.I1) GBA - 1179; (Job.1.S2.I2) GBA - 1177; (Job.1.S2.I3) GBA - 1131; (Job.1.S2.I4) GBA - 1115 : **1154.0** (Job.1.S3.I1) GBA - 1055; (Job.1.S3.I2) GBA - 1963; (Job.1.S3.I3) GBA - 1490; (Job.1.S3.I4) GBA - 1036 : **1272.5** (Job.1.S4.I1) GBA - 1021; (Job.1.S4.I2) GBA - 2219; (Job.1.S4.I3) GBA - 942; (Job.1.S4.I4) GBA - 928 : **981.5**

**Job # 2**

(Job.2.S1.I1) GBA - 2235; (Job.2.S1.I2) GBA - 2221; (Job.2.S1.I3) GBA - 2221; (Job.2.S1.I4) GBA - 2221 : **2221.0** (Job.2.S2.I1) GBA - 2270; (Job.2.S2.I2) GBA - 2270; (Job.2.S2.I3) GBA - 1972; (Job.2.S2.I4) GBA - 2343 : **2270.0** (Job.2.S3.I1) GBA - 2395; (Job.2.S3.I2) GBA - 2205; (Job.2.S3.I3) GBA - 2363; (Job.2.S3.I4) GBA - 2216 : **2289.5** (Job.2.S4.I1) GBA - 997; (Job.2.S4.I2) GBA - 1787; (Job.2.S4.I3) GBA - 1787; (Job.2.S4.I4) GBA - 1707 : **1747.0**

**Job # 3**

(Job.3.S1.I1) GBA - 2351; (Job.3.S1.I2) GBA - 1784; (Job.3.S1.I3) GBA - 2338; (Job.3.S1.I4) GBA - 2463 : **2344.5** (Job.3.S2.I1) GBA - 1954; (Job.3.S2.I2) GBA - 1981; (Job.3.S2.I3) GBA - 1990; (Job.3.S2.I4) GBA - 1961 : **1971.0** (Job.3.S3.I1) GBA - 2074; (Job.3.S3.I2) GBA - 1811; (Job.3.S3.I3) GBA - 2072; (Job.3.S3.I4) GBA - 1847 : **1959.5** (Job.3.S4.I1) GBA - 2013; (Job.3.S4.I2) GBA - 2013; (Job.3.S4.I3) GBA - 1906; (Job.3.S4.I4) GBA - 1906 : **1959.5**

**Job # 4**

(Job.4.S1.I1) GBA - 2676; (Job.4.S1.I2) GBA - 2409; (Job.4.S1.I3) GBA - 2409; (Job.4.S1.I4) GBA - 2639 : **2524.0** (Job.4.S2.I1) GBA - 1836; (Job.4.S2.I2) GBA

- 1840; (Job.4.S2.I3) GBA - 1869; (Job.4.S2.I4) GBA - 1946 : **1854.5** (Job.4.S3.I1) GBA - 2278; (Job.4.S3.I2) GBA - 2481; (Job.4.S3.I3) GBA - 1132; (Job.4.S3.I4) GBA - 2413 : **2345.5** (Job.4.S4.I1) GBA - 1329; (Job.4.S4.I2) GBA - 2221; (Job.4.S4.I3) GBA - 1549; (Job.4.S4.I4) GBA - 1417 : **1483.0**

*Best ranks of each set–island*: 1660 - 1434 - 1754 - 1794 - 1920 - 1115 - 1036 - 928 - 2221 - 1972 - 2205 - 997 - 1784 - 1954 - 1811 - 1906 - 2409 - 1836 - 1132 - 1329 -
*Median*: 1890.25
*Median of best*: 1789.0

### A.2.9 LRRK2

**Job # 0**
(Job.0.S1.I1) LRRK2 - 204; (Job.0.S1.I2) LRRK2 - 182; (Job.0.S1.I3) LRRK2 - 204; (Job.0.S1.I4) LRRK2 - 204 : **204.0** (Job.0.S2.I1) LRRK2 - 641; (Job.0.S2.I2) LRRK2 - 612; (Job.0.S2.I3) LRRK2 - 615; (Job.0.S2.I4) LRRK2 - 616 : **615.5** (Job.0.S3.I1) LRRK2 - 179; (Job.0.S3.I2) LRRK2 - 176; (Job.0.S3.I3) LRRK2 - 179; (Job.0.S3.I4) LRRK2 - 179 : **179.0** (Job.0.S4.I1) LRRK2 - 867; (Job.0.S4.I2) LRRK2 - 870; (Job.0.S4.I3) LRRK2 - 832; (Job.0.S4.I4) LRRK2 - 825 : **849.5**

**Job # 1**
(Job.1.S1.I1) LRRK2 - 251; (Job.1.S1.I2) LRRK2 - 251; (Job.1.S1.I3) LRRK2 - 251; (Job.1.S1.I4) LRRK2 - 326 : **251.0** (Job.1.S2.I1) LRRK2 - 717; (Job.1.S2.I2) LRRK2 - 717; (Job.1.S2.I3) LRRK2 - 724; (Job.1.S2.I4) LRRK2 - 713 : **717.0** (Job.1.S3.I1) LRRK2 - 494; (Job.1.S3.I2) LRRK2 - 931; (Job.1.S3.I3) LRRK2 - 886; (Job.1.S3.I4) LRRK2 - 626 : **756.0** (Job.1.S4.I1) LRRK2 - 479; (Job.1.S4.I2) LRRK2 - 840; (Job.1.S4.I3) LRRK2 - 833; (Job.1.S4.I4) LRRK2 - 485 : **659.0**

**Job # 2**
(Job.2.S1.I1) LRRK2 - 650; (Job.2.S1.I2) LRRK2 - 2408; (Job.2.S1.I3) LRRK2 - 619; (Job.2.S1.I4) LRRK2 - 636 : **643.0** (Job.2.S2.I1) LRRK2 - 743; (Job.2.S2.I2) LRRK2 - 855; (Job.2.S2.I3) LRRK2 - 877; (Job.2.S2.I4) LRRK2 - 733 : **799.0** (Job.2.S3.I1) LRRK2 - 760; (Job.2.S3.I2) LRRK2 - 762; (Job.2.S3.I3) LRRK2 - 755; (Job.2.S3.I4) LRRK2 - 760 : **760.0** (Job.2.S4.I1) LRRK2 - 905; (Job.2.S4.I2) LRRK2 - 906; (Job.2.S4.I3) LRRK2 - 906; (Job.2.S4.I4) LRRK2 - 901 : **905.5**

**Job # 3**

(Job.3.S1.I1) LRRK2 - 439; (Job.3.S1.I2) LRRK2 - 438; (Job.3.S1.I3) LRRK2 - 439; (Job.3.S1.I4) LRRK2 - 489 : **439.0** (Job.3.S2.I1) LRRK2 - 77; (Job.3.S2.I2) LRRK2 - 79; (Job.3.S2.I3) LRRK2 - 78; (Job.3.S2.I4) LRRK2 - 209 : **78.5** (Job.3.S3.I1) LRRK2 - 782; (Job.3.S3.I2) LRRK2 - 528; (Job.3.S3.I3) LRRK2 - 478; (Job.3.S3.I4) LRRK2 - 739 : **633.5** (Job.3.S4.I1) LRRK2 - 255; (Job.3.S4.I2) LRRK2 - 485; (Job.3.S4.I3) LRRK2 - 485; (Job.3.S4.I4) LRRK2 - 255 : **370.0**

**Job # 4**

(Job.4.S1.I1) LRRK2 - 309; (Job.4.S1.I2) LRRK2 - 308; (Job.4.S1.I3) LRRK2 - 221; (Job.4.S1.I4) LRRK2 - 288 : **298.0** (Job.4.S2.I1) LRRK2 - 812; (Job.4.S2.I2) LRRK2 - 808; (Job.4.S2.I3) LRRK2 - 928; (Job.4.S2.I4) LRRK2 - 811 : **811.5** (Job.4.S3.I1) LRRK2 - 721; (Job.4.S3.I2) LRRK2 - 330; (Job.4.S3.I3) LRRK2 - 736; (Job.4.S3.I4) LRRK2 - 246 : **525.5** (Job.4.S4.I1) LRRK2 - 893; (Job.4.S4.I2) LRRK2 - 881; (Job.4.S4.I3) LRRK2 - 881; (Job.4.S4.I4) LRRK2 - 880 : **881.0**

*Best ranks of each set–island*: 182 - 612 - 176 - 825 - 251 - 713 - 494 - 479 - 619 - 733 - 755 - 901 - 438 - 77 - 478 - 255 - 221 - 808 - 246 - 880 -
*Median*: 638.25
*Median of best*: 486.5

### A.2.10 MAOB

**Job # 0**

(Job.0.S1.I1) MAOB - 10; (Job.0.S1.I2) MAOB - 10; (Job.0.S1.I3) MAOB - 10; (Job.0.S1.I4) MAOB - 14 : **10.0** (Job.0.S2.I1) MAOB - 37; (Job.0.S2.I2) MAOB - 37; (Job.0.S2.I3) MAOB - 37; (Job.0.S2.I4) MAOB - 31 : **37.0** (Job.0.S3.I1) MAOB - 24; (Job.0.S3.I2) MAOB - 23; (Job.0.S3.I3) MAOB - 24; (Job.0.S3.I4) MAOB - 28 : **24.0** (Job.0.S4.I1) MAOB - 46; (Job.0.S4.I2) MAOB - 426; (Job.0.S4.I3) MAOB - 426; (Job.0.S4.I4) MAOB - 426 : **426.0**

**Job # 1**

(Job.1.S1.I1) MAOB - 62; (Job.1.S1.I2) MAOB - 648; (Job.1.S1.I3) MAOB - 785; (Job.1.S1.I4) MAOB - 48 : **355.0** (Job.1.S2.I1) MAOB - 81; (Job.1.S2.I2) MAOB - 75; (Job.1.S2.I3) MAOB - 81; (Job.1.S2.I4) MAOB - 67 : **78.0** (Job.1.S3.I1) MAOB - 49; (Job.1.S3.I2) MAOB - 49; (Job.1.S3.I3) MAOB - 28; (Job.1.S3.I4) MAOB - 49 :

**49.0** (Job.1.S4.I1) MAOB - 32; (Job.1.S4.I2) MAOB - 36; (Job.1.S4.I3) MAOB - 30; (Job.1.S4.I4) MAOB - 32 : **32.0**

**Job # 2**

(Job.2.S1.I1) MAOB - 73; (Job.2.S1.I2) MAOB - 73; (Job.2.S1.I3) MAOB - 37; (Job.2.S1.I4) MAOB - 884 : **73.0** (Job.2.S2.I1) MAOB - 13; (Job.2.S2.I2) MAOB - 12; (Job.2.S2.I3) MAOB - 13; (Job.2.S2.I4) MAOB - 13 : **13.0** (Job.2.S3.I1) MAOB - 87; (Job.2.S3.I2) MAOB - 99; (Job.2.S3.I3) MAOB - 84; (Job.2.S3.I4) MAOB - 89 : **88.0** (Job.2.S4.I1) MAOB - 52; (Job.2.S4.I2) MAOB - 52; (Job.2.S4.I3) MAOB - 365; (Job.2.S4.I4) MAOB - 52 : **52.0**

**Job # 3**

(Job.3.S1.I1) MAOB - 19; (Job.3.S1.I2) MAOB - 89; (Job.3.S1.I3) MAOB - 19; (Job.3.S1.I4) MAOB - 235 : **54.0** (Job.3.S2.I1) MAOB - 171; (Job.3.S2.I2) MAOB - 70; (Job.3.S2.I3) MAOB - 109; (Job.3.S2.I4) MAOB - 70 : **89.5** (Job.3.S3.I1) MAOB - 43; (Job.3.S3.I2) MAOB - 29; (Job.3.S3.I3) MAOB - 235; (Job.3.S3.I4) MAOB - 25 : **36.0** (Job.3.S4.I1) MAOB - 32; (Job.3.S4.I2) MAOB - 33; (Job.3.S4.I3) MAOB - 32; (Job.3.S4.I4) MAOB - 34 : **32.5**

**Job # 4**

(Job.4.S1.I1) MAOB - 37; (Job.4.S1.I2) MAOB - 44; (Job.4.S1.I3) MAOB - 54; (Job.4.S1.I4) MAOB - 44 : **44.0** (Job.4.S2.I1) MAOB - 67; (Job.4.S2.I2) MAOB - 74; (Job.4.S2.I3) MAOB - 49; (Job.4.S2.I4) MAOB - 67 : **67.0** (Job.4.S3.I1) MAOB - 103; (Job.4.S3.I2) MAOB - 84; (Job.4.S3.I3) MAOB - 79; (Job.4.S3.I4) MAOB - 96 : **90.0** (Job.4.S4.I1) MAOB - 28; (Job.4.S4.I2) MAOB - 28; (Job.4.S4.I3) MAOB - 28; (Job.4.S4.I4) MAOB - 28 : **28.0**

*Best ranks of each set–island*: 10 - 31 - 23 - 46 - 48 - 67 - 28 - 30 - 37 - 12 - 84 - 52 - 19 - 70 - 25 - 32 - 37 - 49 - 79 - 28 -
*Median*: 50.5
*Median of best*: 34.5

### A.2.11 MAPT

**Job # 0**

(Job.0.S1.I1) MAPT - 299; (Job.0.S1.I2) MAPT - 252; (Job.0.S1.I3) MAPT - 287;

(Job.0.S1.I4) MAPT - 287 : **287.0** (Job.0.S2.I1) MAPT - 510; (Job.0.S2.I2) MAPT - 510; (Job.0.S2.I3) MAPT - 510; (Job.0.S2.I4) MAPT - 510 : **510.0** (Job.0.S3.I1) MAPT - 100; (Job.0.S3.I2) MAPT - 120; (Job.0.S3.I3) MAPT - 100; (Job.0.S3.I4) MAPT - 196 : **110.0** (Job.0.S4.I1) MAPT - 119; (Job.0.S4.I2) MAPT - 119; (Job.0.S4.I3) MAPT - 119; (Job.0.S4.I4) MAPT - 119 : **119.0**

**Job # 1**

(Job.1.S1.I1) MAPT - 215; (Job.1.S1.I2) MAPT - 215; (Job.1.S1.I3) MAPT - 215; (Job.1.S1.I4) MAPT - 50 : **215.0** (Job.1.S2.I1) MAPT - 656; (Job.1.S2.I2) MAPT - 704; (Job.1.S2.I3) MAPT - 701; (Job.1.S2.I4) MAPT - 656 : **678.5** (Job.1.S3.I1) MAPT - 25; (Job.1.S3.I2) MAPT - 550; (Job.1.S3.I3) MAPT - 553; (Job.1.S3.I4) MAPT - 549 : **549.5** (Job.1.S4.I1) MAPT - 125; (Job.1.S4.I2) MAPT - 128; (Job.1.S4.I3) MAPT - 120; (Job.1.S4.I4) MAPT - 317 : **126.5**

**Job # 2**

(Job.2.S1.I1) MAPT - 103; (Job.2.S1.I2) MAPT - 120; (Job.2.S1.I3) MAPT - 108; (Job.2.S1.I4) MAPT - 47 : **105.5** (Job.2.S2.I1) MAPT - 798; (Job.2.S2.I2) MAPT - 801; (Job.2.S2.I3) MAPT - 805; (Job.2.S2.I4) MAPT - 797 : **799.5** (Job.2.S3.I1) MAPT - 880; (Job.2.S3.I2) MAPT - 805; (Job.2.S3.I3) MAPT - 542; (Job.2.S3.I4) MAPT - 865 : **835.0** (Job.2.S4.I1) MAPT - 109; (Job.2.S4.I2) MAPT - 108; (Job.2.S4.I3) MAPT - 109; (Job.2.S4.I4) MAPT - 643 : **109.0**

**Job # 3**

(Job.3.S1.I1) MAPT - 589; (Job.3.S1.I2) MAPT - 307; (Job.3.S1.I3) MAPT - 307; (Job.3.S1.I4) MAPT - 174 : **307.0** (Job.3.S2.I1) MAPT - 493; (Job.3.S2.I2) MAPT - 435; (Job.3.S2.I3) MAPT - 459; (Job.3.S2.I4) MAPT - 478 : **468.5** (Job.3.S3.I1) MAPT - 151; (Job.3.S3.I2) MAPT - 155; (Job.3.S3.I3) MAPT - 160; (Job.3.S3.I4) MAPT - 161 : **157.5** (Job.3.S4.I1) MAPT - 452; (Job.3.S4.I2) MAPT - 452; (Job.3.S4.I3) MAPT - 436; (Job.3.S4.I4) MAPT - 437 : **444.5**

**Job # 4**

(Job.4.S1.I1) MAPT - 493; (Job.4.S1.I2) MAPT - 494; (Job.4.S1.I3) MAPT - 526; (Job.4.S1.I4) MAPT - 759 : **510.0** (Job.4.S2.I1) MAPT - 618; (Job.4.S2.I2) MAPT - 616; (Job.4.S2.I3) MAPT - 326; (Job.4.S2.I4) MAPT - 613 : **614.5** (Job.4.S3.I1) MAPT - 132; (Job.4.S3.I2) MAPT - 138; (Job.4.S3.I3) MAPT - 149; (Job.4.S3.I4) MAPT - 801 : **143.5** (Job.4.S4.I1) MAPT - 764; (Job.4.S4.I2) MAPT - 928; (Job.4.S4.I3)

MAPT - 593; (Job.4.S4.I4) MAPT - 770 : **767.0**

*Best ranks of each set–island*: 252 - 510 - 100 - 119 - 50 - 656 - 25 - 120 - 47 - 797 - 542 - 108 - 174 - 435 - 151 - 436 - 493 - 326 - 132 - 593 -
*Median*: 375.75
*Median of best*: 213.0

### A.2.12 PARK2

**Job # 0**
(Job.0.S1.I1) PARK2 - 4; (Job.0.S1.I2) PARK2 - 4; (Job.0.S1.I3) PARK2 - 4; (Job.0.S1.I4) PARK2 - 4 : **4.0** (Job.0.S2.I1) PARK2 - 5; (Job.0.S2.I2) PARK2 - 5; (Job.0.S2.I3) PARK2 - 5; (Job.0.S2.I4) PARK2 - 4 : **5.0** (Job.0.S3.I1) PARK2 - 33; (Job.0.S3.I2) PARK2 - 2; (Job.0.S3.I3) PARK2 - 4; (Job.0.S3.I4) PARK2 - 4 : **4.0** (Job.0.S4.I1) PARK2 - 4; (Job.0.S4.I2) PARK2 - 4; (Job.0.S4.I3) PARK2 - 1; (Job.0.S4.I4) PARK2 - 4 : **4.0**

**Job # 1**
(Job.1.S1.I1) PARK2 - 5; (Job.1.S1.I2) PARK2 - 5; (Job.1.S1.I3) PARK2 - 5; (Job.1.S1.I4) PARK2 - 35 : **5.0** (Job.1.S2.I1) PARK2 - 4; (Job.1.S2.I2) PARK2 - 4; (Job.1.S2.I3) PARK2 - 4; (Job.1.S2.I4) PARK2 - 4 : **4.0** (Job.1.S3.I1) PARK2 - 2; (Job.1.S3.I2) PARK2 - 2; (Job.1.S3.I3) PARK2 - 4; (Job.1.S3.I4) PARK2 - 5 : **3.0** (Job.1.S4.I1) PARK2 - 5; (Job.1.S4.I2) PARK2 - 6; (Job.1.S4.I3) PARK2 - 6; (Job.1.S4.I4) PARK2 - 5 : **5.5**

**Job # 2**
(Job.2.S1.I1) PARK2 - 12; (Job.2.S1.I2) PARK2 - 13; (Job.2.S1.I3) PARK2 - 20; (Job.2.S1.I4) PARK2 - 9 : **12.5** (Job.2.S2.I1) PARK2 - 5; (Job.2.S2.I2) PARK2 - 5; (Job.2.S2.I3) PARK2 - 5; (Job.2.S2.I4) PARK2 - 5 : **5.0** (Job.2.S3.I1) PARK2 - 5; (Job.2.S3.I2) PARK2 - 5; (Job.2.S3.I3) PARK2 - 5; (Job.2.S3.I4) PARK2 - 5 : **5.0** (Job.2.S4.I1) PARK2 - 25; (Job.2.S4.I2) PARK2 - 25; (Job.2.S4.I3) PARK2 - 25; (Job.2.S4.I4) PARK2 - 22 : **25.0**

**Job # 3**
(Job.3.S1.I1) PARK2 - 5; (Job.3.S1.I2) PARK2 - 5; (Job.3.S1.I3) PARK2 - 4; (Job.3.S1.I4) PARK2 - 5 : **5.0** (Job.3.S2.I1) PARK2 - 4; (Job.3.S2.I2) PARK2 - 4; (Job.3.S2.I3)

PARK2 - 6; (Job.3.S2.I4) PARK2 - 4 : **4.0** (Job.3.S3.I1) PARK2 - 4; (Job.3.S3.I2) PARK2 - 4; (Job.3.S3.I3) PARK2 - 4; (Job.3.S3.I4) PARK2 - 4 : **4.0** (Job.3.S4.I1) PARK2 - 94; (Job.3.S4.I2) PARK2 - 95; (Job.3.S4.I3) PARK2 - 115; (Job.3.S4.I4) PARK2 - 94 : **94.5**

**Job # 4**
(Job.4.S1.I1) PARK2 - 32; (Job.4.S1.I2) PARK2 - 32; (Job.4.S1.I3) PARK2 - 15; (Job.4.S1.I4) PARK2 - 3 : **23.5** (Job.4.S2.I1) PARK2 - 15; (Job.4.S2.I2) PARK2 - 6; (Job.4.S2.I3) PARK2 - 15; (Job.4.S2.I4) PARK2 - 15 : **15.0** (Job.4.S3.I1) PARK2 - 11; (Job.4.S3.I2) PARK2 - 11; (Job.4.S3.I3) PARK2 - 18; (Job.4.S3.I4) PARK2 - 15 : **13.0** (Job.4.S4.I1) PARK2 - 21; (Job.4.S4.I2) PARK2 - 9; (Job.4.S4.I3) PARK2 - 12; (Job.4.S4.I4) PARK2 - 11 : **11.5**

*Best ranks of each set–island*: 4 - 4 - 2 - 1 - 5 - 4 - 2 - 5 - 9 - 5 - 5 - 22 - 4 - 4 - 4 - 94 - 3 - 6 - 11 - 9 -
*Median*: 5.0
*Median of best*: 4.5

### A.2.13 PINK1

**Job # 0**
(Job.0.S1.I1) PINK1 - 2; (Job.0.S1.I2) PINK1 - 1; (Job.0.S1.I3) PINK1 - 2; (Job.0.S1.I4) PINK1 - 2 : **2.0** (Job.0.S2.I1) PINK1 - 823; (Job.0.S2.I2) PINK1 - 5; (Job.0.S2.I3) PINK1 - 812; (Job.0.S2.I4) PINK1 - 868 : **817.5** (Job.0.S3.I1) PINK1 - 4; (Job.0.S3.I2) PINK1 - 4; (Job.0.S3.I3) PINK1 - 5; (Job.0.S3.I4) PINK1 - 4 : **4.0** (Job.0.S4.I1) PINK1 - 6; (Job.0.S4.I2) PINK1 - 5; (Job.0.S4.I3) PINK1 - 8; (Job.0.S4.I4) PINK1 - 5 : **5.5**

**Job # 1**
(Job.1.S1.I1) PINK1 - 6; (Job.1.S1.I2) PINK1 - 6; (Job.1.S1.I3) PINK1 - 4; (Job.1.S1.I4) PINK1 - 6 : **6.0** (Job.1.S2.I1) PINK1 - 37; (Job.1.S2.I2) PINK1 - 30; (Job.1.S2.I3) PINK1 - 11; (Job.1.S2.I4) PINK1 - 35 : **32.5** (Job.1.S3.I1) PINK1 - 4; (Job.1.S3.I2) PINK1 - 4; (Job.1.S3.I3) PINK1 - 4; (Job.1.S3.I4) PINK1 - 9 : **4.0** (Job.1.S4.I1) PINK1 - 4; (Job.1.S4.I2) PINK1 - 4; (Job.1.S4.I3) PINK1 - 4; (Job.1.S4.I4) PINK1 - 57 : **4.0**

**Job # 2**
(Job.2.S1.I1) PINK1 - 64; (Job.2.S1.I2) PINK1 - 63; (Job.2.S1.I3) PINK1 - 64; (Job.2.S1.I4) PINK1 - 61 : **63.5** (Job.2.S2.I1) PINK1 - 41; (Job.2.S2.I2) PINK1 - 33; (Job.2.S2.I3) PINK1 - 41; (Job.2.S2.I4) PINK1 - 36 : **38.5** (Job.2.S3.I1) PINK1 - 1; (Job.2.S3.I2) PINK1 - 4; (Job.2.S3.I3) PINK1 - 4; (Job.2.S3.I4) PINK1 - 4 : **4.0** (Job.2.S4.I1) PINK1 - 4; (Job.2.S4.I2) PINK1 - 4; (Job.2.S4.I3) PINK1 - 4; (Job.2.S4.I4) PINK1 - 4 : **4.0**

**Job # 3**
(Job.3.S1.I1) PINK1 - 6; (Job.3.S1.I2) PINK1 - 6; (Job.3.S1.I3) PINK1 - 6; (Job.3.S1.I4) PINK1 - 4 : **6.0** (Job.3.S2.I1) PINK1 - 5; (Job.3.S2.I2) PINK1 - 5; (Job.3.S2.I3) PINK1 - 29; (Job.3.S2.I4) PINK1 - 29 : **17.0** (Job.3.S3.I1) PINK1 - 13; (Job.3.S3.I2) PINK1 - 38; (Job.3.S3.I3) PINK1 - 38; (Job.3.S3.I4) PINK1 - 34 : **36.0** (Job.3.S4.I1) PINK1 - 6; (Job.3.S4.I2) PINK1 - 6; (Job.3.S4.I3) PINK1 - 4; (Job.3.S4.I4) PINK1 - 4 : **5.0**

**Job # 4**
(Job.4.S1.I1) PINK1 - 5; (Job.4.S1.I2) PINK1 - 5; (Job.4.S1.I3) PINK1 - 5; (Job.4.S1.I4) PINK1 - 5 : **5.0** (Job.4.S2.I1) PINK1 - 110; (Job.4.S2.I2) PINK1 - 111; (Job.4.S2.I3) PINK1 - 90; (Job.4.S2.I4) PINK1 - 5 : **100.0** (Job.4.S3.I1) PINK1 - 5; (Job.4.S3.I2) PINK1 - 5; (Job.4.S3.I3) PINK1 - 4; (Job.4.S3.I4) PINK1 - 5 : **5.0** (Job.4.S4.I1) PINK1 - 60; (Job.4.S4.I2) PINK1 - 62; (Job.4.S4.I3) PINK1 - 64; (Job.4.S4.I4) PINK1 - 62 : **62.0**

*Best ranks of each set–island*: 1 - 5 - 4 - 5 - 4 - 11 - 4 - 4 - 61 - 33 - 1 - 4 - 4 - 5 - 13 - 4 - 5 - 5 - 4 - 60 -
*Median*: 5.75
*Median of best*: 4.5

### A.2.14 PON1

**Job # 0**
(Job.0.S1.I1) PON1 - 34; (Job.0.S1.I2) PON1 - 36; (Job.0.S1.I3) PON1 - 34; (Job.0.S1.I4) PON1 - 35 : **34.5** (Job.0.S2.I1) PON1 - 36; (Job.0.S2.I2) PON1 - 32; (Job.0.S2.I3) PON1 - 32; (Job.0.S2.I4) PON1 - 62 : **34.0** (Job.0.S3.I1) PON1 - 790; (Job.0.S3.I2) PON1 - 670; (Job.0.S3.I3) PON1 - 682; (Job.0.S3.I4) PON1 - 847 : **736.0** (Job.0.S4.I1)

PON1 - 52; (Job.0.S4.I2) PON1 - 39; (Job.0.S4.I3) PON1 - 39; (Job.0.S4.I4) PON1 - 52 : **45.5**

**Job # 1**

(Job.1.S1.I1) PON1 - 40; (Job.1.S1.I2) PON1 - 55; (Job.1.S1.I3) PON1 - 82; (Job.1.S1.I4) PON1 - 39 : **47.5** (Job.1.S2.I1) PON1 - 342; (Job.1.S2.I2) PON1 - 342; (Job.1.S2.I3) PON1 - 274; (Job.1.S2.I4) PON1 - 305 : **323.5** (Job.1.S3.I1) PON1 - 37; (Job.1.S3.I2) PON1 - 28; (Job.1.S3.I3) PON1 - 31; (Job.1.S3.I4) PON1 - 30 : **30.5** (Job.1.S4.I1) PON1 - 34; (Job.1.S4.I2) PON1 - 34; (Job.1.S4.I3) PON1 - 47; (Job.1.S4.I4) PON1 - 35 : **34.5**

**Job # 2**

(Job.2.S1.I1) PON1 - 56; (Job.2.S1.I2) PON1 - 49; (Job.2.S1.I3) PON1 - 52; (Job.2.S1.I4) PON1 - 30 : **50.5** (Job.2.S2.I1) PON1 - 64; (Job.2.S2.I2) PON1 - 28; (Job.2.S2.I3) PON1 - 101; (Job.2.S2.I4) PON1 - 64 : **64.0** (Job.2.S3.I1) PON1 - 81; (Job.2.S3.I2) PON1 - 65; (Job.2.S3.I3) PON1 - 63; (Job.2.S3.I4) PON1 - 69 : **67.0** (Job.2.S4.I1) PON1 - 26; (Job.2.S4.I2) PON1 - 99; (Job.2.S4.I3) PON1 - 98; (Job.2.S4.I4) PON1 - 79 : **88.5**

**Job # 3**

(Job.3.S1.I1) PON1 - 366; (Job.3.S1.I2) PON1 - 136; (Job.3.S1.I3) PON1 - 380; (Job.3.S1.I4) PON1 - 428 : **373.0** (Job.3.S2.I1) PON1 - 36; (Job.3.S2.I2) PON1 - 37; (Job.3.S2.I3) PON1 - 36; (Job.3.S2.I4) PON1 - 33 : **36.0** (Job.3.S3.I1) PON1 - 35; (Job.3.S3.I2) PON1 - 36; (Job.3.S3.I3) PON1 - 36; (Job.3.S3.I4) PON1 - 28 : **35.5** (Job.3.S4.I1) PON1 - 36; (Job.3.S4.I2) PON1 - 36; (Job.3.S4.I3) PON1 - 36; (Job.3.S4.I4) PON1 - 36 : **36.0**

**Job # 4**

(Job.4.S1.I1) PON1 - 76; (Job.4.S1.I2) PON1 - 77; (Job.4.S1.I3) PON1 - 77; (Job.4.S1.I4) PON1 - 66 : **76.5** (Job.4.S2.I1) PON1 - 65; (Job.4.S2.I2) PON1 - 66; (Job.4.S2.I3) PON1 - 77; (Job.4.S2.I4) PON1 - 47 : **65.5** (Job.4.S3.I1) PON1 - 71; (Job.4.S3.I2) PON1 - 80; (Job.4.S3.I3) PON1 - 72; (Job.4.S3.I4) PON1 - 49 : **71.5** (Job.4.S4.I1) PON1 - 35; (Job.4.S4.I2) PON1 - 31; (Job.4.S4.I3) PON1 - 35; (Job.4.S4.I4) PON1 - 35 : **35.0**

*Best ranks of each set–island*: 34 - 32 - 670 - 39 - 39 - 274 - 28 - 34 - 30 - 28 -

63 - 26 - 136 - 33 - 28 - 36 - 66 - 47 - 49 - 31 -
*Median*: 49.0
*Median of best*: 35.0

### A.2.15 SNCA

**Job # 0**
(Job.0.S1.I1) SNCA - 191; (Job.0.S1.I2) SNCA - 4; (Job.0.S1.I3) SNCA - 2; (Job.0.S1.I4) SNCA - 5 : **4.5** (Job.0.S2.I1) SNCA - 357; (Job.0.S2.I2) SNCA - 469; (Job.0.S2.I3) SNCA - 480; (Job.0.S2.I4) SNCA - 468 : **468.5** (Job.0.S3.I1) SNCA - 17; (Job.0.S3.I2) SNCA - 17; (Job.0.S3.I3) SNCA - 18; (Job.0.S3.I4) SNCA - 17 : **17.0** (Job.0.S4.I1) SNCA - 4; (Job.0.S4.I2) SNCA - 4; (Job.0.S4.I3) SNCA - 4; (Job.0.S4.I4) SNCA - 2693 : **4.0**

**Job # 1**
(Job.1.S1.I1) SNCA - 5; (Job.1.S1.I2) SNCA - 6; (Job.1.S1.I3) SNCA - 6; (Job.1.S1.I4) SNCA - 5 : **5.5** (Job.1.S2.I1) SNCA - 83; (Job.1.S2.I2) SNCA - 23; (Job.1.S2.I3) SNCA - 66; (Job.1.S2.I4) SNCA - 66 : **66.0** (Job.1.S3.I1) SNCA - 4; (Job.1.S3.I2) SNCA - 3; (Job.1.S3.I3) SNCA - 4; (Job.1.S3.I4) SNCA - 3 : **3.5** (Job.1.S4.I1) SNCA - 4; (Job.1.S4.I2) SNCA - 4; (Job.1.S4.I3) SNCA - 4; (Job.1.S4.I4) SNCA - 4 : **4.0**

**Job # 2**
(Job.2.S1.I1) SNCA - 22; (Job.2.S1.I2) SNCA - 78; (Job.2.S1.I3) SNCA - 109; (Job.2.S1.I4) SNCA - 24 : **51.0** (Job.2.S2.I1) SNCA - 17; (Job.2.S2.I2) SNCA - 17; (Job.2.S2.I3) SNCA - 24; (Job.2.S2.I4) SNCA - 16 : **17.0** (Job.2.S3.I1) SNCA - 3; (Job.2.S3.I2) SNCA - 4; (Job.2.S3.I3) SNCA - 3; (Job.2.S3.I4) SNCA - 4 : **3.5** (Job.2.S4.I1) SNCA - 11; (Job.2.S4.I2) SNCA - 136; (Job.2.S4.I3) SNCA - 23; (Job.2.S4.I4) SNCA - 107 : **65.0**

**Job # 3**
(Job.3.S1.I1) SNCA - 901; (Job.3.S1.I2) SNCA - 904; (Job.3.S1.I3) SNCA - 901; (Job.3.S1.I4) SNCA - 901 : **901.0** (Job.3.S2.I1) SNCA - 3; (Job.3.S2.I2) SNCA - 2; (Job.3.S2.I3) SNCA - 3; (Job.3.S2.I4) SNCA - 3 : **3.0** (Job.3.S3.I1) SNCA - 136; (Job.3.S3.I2) SNCA - 196; (Job.3.S3.I3) SNCA - 874; (Job.3.S3.I4) SNCA - 3 : **166.0** (Job.3.S4.I1) SNCA - 395; (Job.3.S4.I2) SNCA - 424; (Job.3.S4.I3) SNCA - 428; (Job.3.S4.I4) SNCA - 395 : **409.5**

**Job # 4**

(Job.4.S1.I1) SNCA - 43; (Job.4.S1.I2) SNCA - 403; (Job.4.S1.I3) SNCA - 337; (Job.4.S1.I4) SNCA - 25 : **190.0** (Job.4.S2.I1) SNCA - 497; (Job.4.S2.I2) SNCA - 453; (Job.4.S2.I3) SNCA - 505; (Job.4.S2.I4) SNCA - 3 : **475.0** (Job.4.S3.I1) SNCA - 40; (Job.4.S3.I2) SNCA - 24; (Job.4.S3.I3) SNCA - 15; (Job.4.S3.I4) SNCA - 25 : **24.5** (Job.4.S4.I1) SNCA - 3; (Job.4.S4.I2) SNCA - 3; (Job.4.S4.I3) SNCA - 4; (Job.4.S4.I4) SNCA - 3 : **3.0**

*Best ranks of each set–island*: 2 - 357 - 17 - 4 - 5 - 23 - 3 - 4 - 22 - 16 - 3 - 11 - 901 - 2 - 3 - 395 - 25 - 3 - 15 - 3 -
*Median*: 20.75
*Median of best*: 8.0